

FLÁVIO DE ALMEIDA E SILVA

**REDES NEURAIS HIERÁRQUICAS PARA
IMPLEMENTAÇÃO DE COMPORTAMENTOS EM AGENTES
AUTÔNOMOS**

**Florianópolis - SC
Fevereiro/2001**

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA
COMPUTAÇÃO**

Flávio de Almeida e Silva

**REDES NEURAIS HIERÁRQUICAS PARA
IMPLEMENTAÇÃO DE COMPORTAMENTOS EM AGENTES
AUTÔNOMOS**

Dissertação submetida à Universidade Federal de Santa Catarina como parte dos
requisitos para a obtenção do grau de Mestre em Ciência da Computação

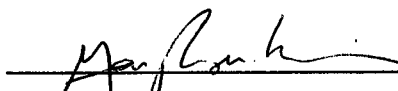
Orientador: Prof. Mauro Roisenberg, Dr.

Florianópolis, fevereiro de 2001

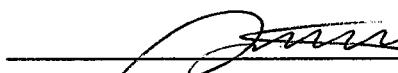
REDES NEURAIS HIERÁRQUICAS PARA IMPLEMENTAÇÃO DE COMPORTAMENTOS EM AGENTES AUTÔNOMOS

Flávio de Almeida e Silva

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Ciência da Computação Área de Concentração: Inteligência Artificial e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

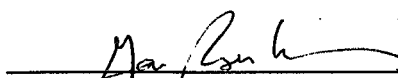


Prof. Dr. Mauro Roisenberg, Orientador




Prof. Dr. Fernando A. Gauthier
Coord. da Pós-Graduação em Ciência da Computação

Banca Examinadora




Prof. Dr. Mauro Roisenberg, Orientador



Prof. DSc. Jorge Muniz Barreto, Co-orientador



Prof. Dr. Edson Roberto De Pieri



Prof. M. Jovelino Falqueto



Prof. Dr. João Bosco da Mota Alves

EPÍGRAFE

“Quando tiver algum problema faça alguma coisa! Se não puder passar por cima, passe por baixo, passe através, dê a volta, vá pela direita, vá pela esquerda. Se não puder obter o material certo, vá procurá-lo.

Se não puder encontrá-lo, substitua-o.

Se não puder substituí-lo, improvise.

Se não puder improvisar, inove.

Mas acima de tudo, faça alguma coisa!!

Há dois gêneros de pessoas que nunca chegam a lugar nenhum: as que não querem fazer nada e as que só inventam desculpas”.

(In: A Essência da Verdade,
Coleção Pensamentos de Sabedoria)

“A história tem demonstrado que os mais notáveis vencedores normalmente encontram obstáculos dolorosos antes de triunfarem. Venceram porque se negaram a serem desencorajados por suas derrotas.”

B. C. Forbes

DEDICATÓRIA

Dedico este trabalho à minha Mãe, Abigail de Carvalho e Silva (In Memoriam), por tudo que me ensinou e pelo exemplo de pessoa que era. Ao meu Pai, João de Almeida e Silva, por tudo que foi e ainda é em minha vida. Pelos seus ensinamentos, pela sua dignidade e caráter. Aos meus irmãos e amigos por toda força e ajuda que me deram. Dedico também ao meu orientador Mauro Roisenberg e co-orientador Jorge Muniz Barreto, por toda paciência e conhecimentos passados, antes e durante a realização deste trabalho. Uma dedicação especial a Deus, pela minha saúde e ajuda nas horas difíceis.

AGRADECIMENTOS

Agradeço ao meu Pai e Irmãos pela força.
Ao meu Orientador e Co-orientador.
Aos meus amigos pelo incentivo.
Agradeço a todos que de forma direta e indireta
contribuíram para a realização deste trabalho.
Em especial a Deus, por não deixar me abater
diante das dificuldades.

PUBLICAÇÕES

SILVA, Flávio de Almeida; ROISENBERG, Mauro; BARRETO, Jorge Muniz. Redes Neurais Hierárquicas para Implementação de Comportamentos em Agentes Autônomos. Congresso Brasileiro de Automática – CBA 2000.

SUMÁRIO

LISTA DE FIGURAS.....	X
LISTA DE TABELAS.....	XI
LISTA DE ABREVIATURAS.....	XII
RESUMO.....	XIII
ABSTRACT.....	XIV
1 INTRODUÇÃO.....	1
1.1 MOTIVAÇÃO.....	2
1.2 OBJETIVOS.....	5
1.2.1 Objetivo Principal.....	5
1.2.2 Objetivos Secundários.....	5
1.3 ORGANIZAÇÃO DO TEXTO.....	6
2 A EVOLUÇÃO BIOLÓGICA, ANATOMIA E FISIOLOGIA DO SISTEMA NERVOSO E COMPORTAMENTO ANIMAL.....	9
2.1 EVOLUÇÃO.....	9
2.2 ANATOMIA E FISIOLOGIA DO SISTEMA NERVOSO.....	13
2.2.1 Anatomia.....	13
2.2.2 Fisiologia.....	18
2.3 REDES NEURAIS MODULARES.....	27
2.4 NOÇÕES DE ETOLOGIA.....	30
2.4.1 Classes de Comportamentos.....	32
2.4.1.1 Comportamentos Reflexivos.....	32
2.4.1.2 Comportamentos Reativos.....	33
3 ALTERNATIVAS DE IMPLEMENTAÇÃO PARA AGENTES AUTÔNOMOS.....	34
4 CONTROLE HIERÁRQUICO DOS COMPORTAMENTOS DE UM AGENTE AUTÔNOMO.....	45
4.1 COMPORTAMENTOS REATIVOS E REFLEXIVOS.....	46
4.1.1 Rede Neural Artificial Hierárquica.....	46
4.1.2 Prioridade de Comportamento.....	51
4.1.3 Descrição dos Comportamentos Implementados.....	54
4.1.3.1 Rede Sensoria de Distância e de Luminosidade.....	55
4.1.3.2 Rede de Controle dos Comportamentos Reflexivos e Reativos.....	56
4.1.3.3 Rede Motora de Distância e de Luminosidade.....	58
4.2 COMPORTAMENTOS REATIVOS, REFLEXIVOS E CONTROLE DE ENERGIA.....	61

4.2.1 Alterações na Estrutura da Hierarquia	61
4.2.2 Descrição dos Novos Comportamentos Implementados	64
4.2.2.1 Rede Sensória Seguir Parede	64
4.2.2.2 Rede Sensória de Distância e de Energia	65
4.2.2.3 Controle de Energia	66
4.2.2.4 Rede de Controle Seguir Parede	68
4.2.2.5 Rede de Controle de Energia	71
4.2.2.6 Rede Motora de Distância da Nova Arquitetura	74
4.2.2.7 Rede Motora de Energia	76
4.2.3 Hierarquia dos Autômatos de Estado Finito	77
5 TREINAMENTO DAS REDES DA ARQUITETURA PARA CONTROLE DOS COMPORTAMENTOS REFLEXIVOS E REATIVOS	79
5.1.1 Definições e o Aprendizado da Rede Hierárquica para Controle dos Comportamentos Reflexivos e Reativos	79
5.1.1.1 Rede Sensória de Distância	79
5.1.1.2 Rede Sensória de Luz	83
5.1.1.3 Rede Motora de Distância – Comportamento Reflexivo	86
5.1.1.4 Rede Motora de Luminosidade – Comportamento Reflexivo	89
5.1.1.5 Rede de Controle – Comportamentos Reativos	91
5.1.2 Análise dos Resultados	92
5.1.3 Definições e Aprendizado da Nova Arquitetura para Controle dos Comportamentos Reflexivos, Reativos e Controle de Energia	97
5.1.3.1 Conjunto de Treinamento para a Rede Sensória Seguir Parede	97
5.1.3.2 Conjunto de Treinamento das Redes Sensórias de Distância e de Energia	99
5.1.3.3 Conjunto de Treinamento da Rede de Controle Seguir Parede	100
5.1.3.4 Conjunto de Treinamento da Rede de Controle de Energia	101
5.1.3.5 Conjunto de Treinamento da Rede Motora de Distância	102
5.1.3.6 Conjunto de treinamento da Rede Motora de Energia	103
5.1.4 Sequência dos Experimentos	105
5.1.5 Análise dos Resultados	107
6 CONCLUSÃO	109
6.1 PROPOSTA PARA TRABALHOS FUTUROS	113
REFERÊNCIAS BIBLIOGRÁFICAS	115
ANEXO A - O AMBIENTE DE DESENVOLVIMENTO: “KHEPERA SIMULADOR”	122
A.1 DESCRIÇÃO DO MUNDO	124
A.2 DESCRIÇÃO DO ROBÔ	125
A.2.1 Apresentação	125
A.2.2 Os Motores	125
A.2.3 Os sensores	126
A.2.4 Operando o Robô	126
A.2.5 Programação	126
A.2.5.1 Os códigos Fonte do Simulador	126

A.2.5.2 Arquivos USER	127
A.2.5.3 Arquivos Fonte do Controlador	127
A.2.5.4 Informações Criadas por Usuários.....	128
A.2.5.5 Exemplos	128
A.2.5.6 Programando um Controlador do Robô de Khepera	129

LISTA DE FIGURAS

Figura 2.1: Sistema Nervoso do Cavalo	15
Figura 2.2: Estrutura de um Neurônio.	16
Figura 2.3: Estrutura do sistema nervoso da Anêmona-do-mar.	20
Figura 2.4: Locomoção da minhoca	21
Figura 2.5: Estrutura das conexões nervosas em uma minhoca.	22
Figura 3.1: Arquitetura em camadas de comportamentos da “Subsumption Architecture”.....	36
Figura 3.2: Esquema Inibidor/Supressor para camada de comportamento.	37
Figura 3.3: Criatura “Paranóico”	40
Figura 3.4: “Inhumane” criatura Inumano.	43
Figura 4.1: Arquitetura NASREM - NASA/NBS standard reference model- Journal of the Brazilian Computer Society.....	47
Figura 4.2: Arquitetura TCA (Task Control Architecture)- Journal of the Brazilian Computer Society.....	48
Figura 4.3: Rede Neural Hierárquica proposta para o controle dos comportamentos.	51
Figura 4.4: Prioridade de comportamento	53
Figura 4.5: Redes Sensórias de Distância e de Luz.	55
Figura 4.6: Rede de Controle – Hierarquia dos Comportamentos Reflexivos e Reativos	56
Figura 4.7: Representação da Rede de Controle Através de um AEF	57
Figura 4.8: Redes Motoras de Distância e de Luz	60
Figura 4.9: Representação Gráfica da Rede Neural Hierárquica.....	60
Figura 4.10: Representação Gráfica da Nova Rede Neural Hierárquica.	62
Figura 4.11: Rede Neural Direta Sensória – Seguir Parede.....	65
Figura 4.12: Redes Neurais Diretas – Sensória de Distância e de Energia	66
Figura 4.13: Gráfico da Função de Saída do Controle de Energia	67
Figura 4.14: Autômato de Estado Finito do Controle de Energia	68
Figura 4.15: Rede Neural Recorrente de Controle – Seguir Parede	70
Figura 4.16: Autômato de Estado Finito (AEF) da Rede de Controle Seguir Parede. ...	70
Figura 4.17: Autômato de Estado Finito (AEF) da Rede de Controle de Energia.	73
Figura 4.18: Rede Neural Recorrente de Controle - Energia	74
Figura 4.19: Rede Neural Direta Motora – Distância	75
Figura 4.20: Rede Neural Direta Motora - Energia	77
Figura 4.21: Separação em níveis da Nova Hierarquia	77
Figura 4.22: Hierarquia dos Autômatos de Estado Finito (AEF)	78
Figura 5.1: Fase I, teste sem obstáculos ou luz.....	94
Figura 5.2: Fase II, teste com obstáculos.....	94
Figura 5.3: Fase III, teste para encontrar luz.	95
Figura 5.4: Fase IV, teste com obstáculo e foco de luz.	96
Figura 5.5: Fase V, teste com foco de luz atrás de um obstáculo	96
Figura A.1: Representação do Deus-Sol Re através da forma “Khepera” (Khepra, Khepri, Khopri).....	122
Figura A.2: Ambiente do robô simulado – Robô de Khepera	124
Figura A.3: Khepera (5 cm diâmetro) e sua contraparte simulada	125

LISTA DE TABELAS

Tabela 5.1: Conjunto de treinamento da Rede Neural Direta – Sensores de Distância..	81
Tabela 5.2: Conjunto de exemplos – Entradas e saídas.....	82
Tabela 5.3: Conjunto de treinamento para a Rede Neural Direta – Sensores de Luminosidade	84
Tabela 5.4: Conjunto de exemplos – Entradas e Saídas	85
Tabela 5.5: Conjunto de treinamento para a Rede Neural Direta – KhepMotD.....	87
Tabela 5.6: Exemplo do conjunto de entradas para a rede KhepMotD.	88
Tabela 5.7: Conjunto de Exemplos – Rede KhepMotD.	88
Tabela 5.8: Conjunto de Entradas – Rede KhepMotL.	89
Tabela 5.9: Exemplo do conjunto de entradas – Rede KhepMotL.	90
Tabela 5.10: Conjunto de exemplos – Rede KhepMotL.	90
Tabela 5.11: Conjunto de entradas – KhepRec.ent.....	92
Tabela 5.12: Conjunto de entradas – KhepRec.dat.....	92
Tabela 5.13: Conjunto de Treinamento da Rede Sensória Seguir Parede.	98
Tabela 5.14: Conjunto de treinamento da Rede Sensória de Distância.....	99
Tabela 5.15: Conjunto de treinamento da Rede Neural Sensória de Energia.....	99
Tabela 5.16: Conjunto de Treinamento da Rede de Controle Seguir Parede	100
Tabela 5.17: Conjunto de Treinamento da Rede de Controle de Energia	101
Tabela 5.18: Conjunto de Treinamento da Rede Motora de Distância.....	102
Tabela 5.19: Conjunto de Treinamento da Rede Motora de Energia	104

LISTA DE ABREVIATURAS

AA	– Agentes Autônomos
AG	– Algoritmos Genéticos
IA	– Inteligência Artificial
AEF	– Autômato de Estados Finitos
RNA's	– Redes Neurais Artificiais
RNAH	– Redes Neurais Artificiais Hierárquicas
SNC	– Sistema Nervoso Central
CNS	– Central Nervous System

RESUMO

Ao utilizar-se os mecanismos naturais como fonte de inspiração para a implementação de comportamentos em Agentes Autônomos (AA's), pode-se constatar a existência de classes de comportamentos cada vez mais complexas. Estes comportamentos parecem estar relacionados com o grau de complexidade da organização do sistema nervoso, isto é, considerando-se uma estrutura hierárquica, comportamentos simples são gerados por sistemas nervosos elementares, por outro lado, comportamentos complexos emergem pela ação de níveis superiores do sistema nervoso, os quais controlam, ordenam e selecionam os comportamentos mais simples. Este trabalho visa mostrar a existência de uma hierarquia nas redes neurais biológicas, como também, serão apresentadas alternativas para a implementação de AA's e o funcionamento do simulador, no qual será implementada a Rede Neural Hierárquica para os testes. Demonstrar-se-á, neste trabalho, a possibilidade da utilização da inspiração na natureza, criando-se uma estrutura hierárquica de diferentes topologias de redes neurais artificiais, como o equivalente computacional do sistema nervoso central (SNC), para elevar o grau de autonomia e utilidade dos AA's.

ABSTRACT

When using the natural mechanisms as inspiration source for the implementation of behaviors in Autonomous Agents (AA's), it can be verified that classes of behaviors exist more and more complex. These behaviors seem to be related with the degree of complexity of the organization of the nervous system, that is, being considered a hierarchical structure, simple behaviors are generated by elementary nervous systems, on the other hand, complex behaviors emerge for the action of superior levels of the nervous system that control, order and select the simplest behaviors. This work presents the hierarchy that exists in the biological neural networks. It will also presents alternatives for the implementation of AA's, as well as, the presentation of the simulator where the work will be developed. It will be demonstrated, in this work, that it is possible to use this inspiration in the Nature, creating a hierarchical structure of different topologies of artificial neural networks as the computational equivalent of the central nervous system (CNS). Being gotten, this way, to elevate the autonomy degree and usefulness of AA's.

1 INTRODUÇÃO

Desde há muito tempo, o Homem tem procurado criar artefatos que o substitua ou auxilie nas tarefas estressantes, perigosas e repetitivas. Em quase todas essas "criações", os seres vivos têm servido como inspiração, seja nas formas, seja nos comportamentos. Parece que insistentemente procura-se reproduzir os seres vivos, utilizando elementos "mortos", como fios, metais, engrenagens, polias, etc.

Observando pelo aspecto comportamental, nota-se que todas as ações dos seres vivos, dos quais foram retiradas as inspirações para o desenvolvimento destes artefatos, são direcionadas aos objetivos de sobrevivência e de reprodução, para que os mesmos comportamentos possam então ser transmitidos para as gerações futuras. Na busca por atingir esses objetivos, os seres vivos necessitam interagir da melhor maneira possível com o ambiente que os cerca. Esta interação pode ser inata ou ainda ser modificada e aperfeiçoada ao longo da existência através do aprendizado, utilizando um atributo que se convencionou chamar de "inteligência". Voltando aos artefatos artificiais, nos dias atuais, o Homem tem utilizado os recursos computacionais para o desenvolvimento de "seres artificiais inteligentes", capazes de "sobreviver" e interagir em vários ambientes auxiliando-o ou mesmo substituindo-o nas mais diversas tarefas. A esses seres artificiais inteligentes dá-se o nome de Agentes Autônomos (AA's).

Os agentes podem "sobreviver" e operar em um ambiente primordialmente "virtual", como por exemplo, a Internet. Chamam-se, os agentes que operam nestes ambientes, de "agentes em software". Por outro lado, os agentes podem ser artefatos físicos e operar em ambientes reais, como fábricas ou vulcões e são chamados de "agentes em hardware" ou "sistemas móveis", cada um com sua função. A evolução das pesquisas sobre AA's está em ascendência, juntamente ou graças à franca evolução da tecnologia, uma vez que estão intimamente ligados. Neste trabalho, abordar-se-á o agente em hardware, pois sobre ele se torna mais evidente a implementação de comportamentos normalmente encontrados nos seres vivos. Nesta categoria tem-se os robôs móveis de todos os tipos - em forma de tambores com rodas para sua locomoção, em forma de animais quadrúpedes ou bípedes, os semelhantes a insetos com várias

“sapatas” e, por fim, os que não se assemelham a animais no formato, mas procuram imitá-los no comportamento.

1.1 Motivação

Enquanto alguns pesquisadores procuravam alternativas algébricas para contornar problemas de colisão entre objetos tridimensionais (CANNY, 1984), outros, na mesma década e até hoje, intensificaram o trabalho procurando aumentar o grau de inteligência e autonomia de Agentes Autônomos (AA's), como podemos ver em (ANDERSON & DONATH, 1991), (BEER et al, 1991), (BRETON et al, 1997), (BROOKS, 1986, 1990ab), (CRISTIAN, 1989), (MATARIC, 1994), (MAES, 1990, 1995), (NABBE, 1998), (NETO & GOES, 1997) (ROISENBERG, 1996), (SARACHIK, 1989), (URBAN et al, 1997). Além de sistemas especialistas, técnicas relativamente recentes de inteligência artificial, tais quais algoritmos genéticos e redes neurais artificiais (FREEMAN & SKAPURA, 1992), e até mesmo, abordagens híbridas destas técnicas estão sendo utilizadas (ANDERSON & DONATH, 1991), (BEER et al, 1991), (SMIEJA, 1992), (BRASIL et al, 1997), (BROOKS, 1986, 1990ab), (MAES, 1990), (MORIARTY & MIKKULAINEN, 1998), (NABBE, 1998), (OLIVEIRA, 1997), (ROISENBERG, 1996, 1999). Apesar de grandes progressos alcançados, ainda não existe uma solução ótima bem estabelecida, e mesmo soluções satisfatórias raramente têm sido apresentadas.

Nos últimos anos têm-se visto implementações que apresentam um grande apelo popular, destinadas a demonstrar ao grande público as possíveis potencialidades da robótica, tais como AA's que sobem cordas e escadas, que repetem os gestos físicos dos humanos ou ainda AA's capazes de penetrar em escombros e ruínas de edificações à procura de sobreviventes. Também entram nestas definições os Veículos de Transporte Autônomos (BORGES & LIMA, 1997). Apesar de sua importância, a característica destas implementações é apresentar um repertório de comportamentos restritos (ou definidos) e com um grau de autonomia limitado. Isto leva a procurar alternativas plausíveis para o aumento do repertório, assim como, do grau de autonomia. Existe no animal um mecanismo inteligente de avaliação de situações e seleção de

comportamentos, sistema nervoso central (SNC), que determina qual comportamento deve emergir para manter sua sobrevivência no mundo. Sendo assim, este mecanismo pode ser a inspiração para a implementação de comportamentos. Dentro do SNC temos módulos de processamento que trabalham sozinhos para processar vários tipos de informações ou módulos que trabalham em conjunto para processar, paralelamente, uma informação apenas.

Esta organização modular dá-nos inspiração para o desenvolvimento de trabalhos computacionais, pois foi um dos modos que a natureza encontrou para dar inteligência aos seres vivos. A proposta deste trabalho está baseada na estrutura horizontal e vertical da organização modular, onde existem neurônios formando linhas (horizontal) e colunas (vertical), e como esta estrutura interage na emergência de comportamentos. Os detalhes da proposta serão apresentados no capítulo referente ao trabalho em si.

Podemos perceber que muitos dos atos dos seres vivos são previamente calculados, pois já haviam passado por situação parecida ou foi-lhes ensinado a maneira de agir diante dela. Então, podemos dizer que o SNC aprendeu e guardou num local do cérebro, para reaproveitar no momento certo, por ter sido considerado algo de suma importância para a sua “sobrevivência”.

Na área de implementação de AA's, um grande progresso pode ser alcançado se for possível tornar o AA um ser capaz de perceber a atitude a ser tomada diante de uma situação já encontrada ou não, porque em sua “memória” foi armazenado algo parecido. Portanto, ele deverá ser capaz de escolher que “caminho tomar” ou qual atitude deve ser tomada. Como se trata de ações a serem tomadas e não apenas uma ação, deve existir uma entidade que faça a seleção e a coordenação dessas ações, permitindo que a mais adequada ao momento se manifeste.

Um exemplo seria dar um comando para o AA pegar um objeto em uma sala, mas no meio do caminho ele “percebe” que as suas baterias estão quase acabando. Qual atitude ele deve tomar? Continuar andando até o objeto ou primeiro recarregar suas baterias? Isto é um tipo de comportamento a ser tratado, e tem-se que estudar quais comportamentos são priorizados, pois sendo facilmente perceptível a existência da

hierarquia no SNC dos animais, pode-se também criar uma hierarquia semelhante nos AA's.

Pode-se perceber que nos animais os comportamentos emergem rumo a dois objetivos, a sobrevivência e a procriação, ao mesmo tempo em que tentam desempenhar seu papel em vários ambientes, desviando de obstáculos que possam atrapalhá-los nestas tarefas. Exemplos para estes comportamentos são: se um animal está com fome, irá procurar comida, mas se entre ele e a comida estiver um predador, irá procurar contornar o problema mesmo que isto signifique deixar a comida encontrada para trás e ir em busca de outra, pois, enfrentar o predador seria um risco muito grande para atingir o objetivo, o alimento. Isto mostra que o comportamento "sobreviver" teve uma maior influência sobre o comportamento "alimentar-se".

Um ponto, que parece comum a todos os comportamentos aqui apresentados, é a existência de uma "coordenação" e "hierarquia" sobre estes comportamentos. Ou seja, existem diversas ocasiões em que o procedimento correto deve ser selecionado de um "repertório" de comportamentos disponíveis e, além disso, eventualmente vários destes comportamentos são antagônicos (fugir do perigo e buscar comida, por exemplo). Deve haver então uma "entidade" capaz de organizar, selecionar e coordenar condutas mais elementares, fazendo emergir um comportamento mais complexo e adequado à operação do ser vivo.

Nesta busca por AA's, com repertórios de comportamentos e grau de autonomia cada vez maiores, tem-se buscado na inspiração biológica (HILLER, 1993), e em especial, na advinda do comportamento animal (ROISENBERG, 1996), ferramentas para a implementação de comportamentos em AA's, ou seja, o desenvolvimento de AA's que apresentem condutas que se assemelham às de animais. Se um animal, um antílope, estiver bebendo água e um predador, um leão, se aproximar para abatê-lo, o antílope sairá em disparada para não ser morto. O comportamento adotado pelo antílope foi de parar de saciar sua vontade, que também era importante para sua sobrevivência, e correr do seu predador. O exemplo anterior tenta mostrar a existência, no animal, de um mecanismo inteligente de avaliação de situações e seleção de comportamentos que nesta situação escolherá um dos comportamentos que permitirá sua sobrevivência. Também

existe um mecanismo de aprendizado associativo de modo que, se um rato comer algo que lhe faça mal, este evitará comer novamente o mesmo alimento.

Com isto, procura-se dar uma contribuição nesta busca, usando especialmente o estudo do sistema nervoso central e como ele se organiza nos seres vivos para a geração, seleção e coordenação de comportamentos capazes de garantir sua autonomia nos seus ambientes.

Faz-se necessário, para um melhor entendimento, uma descrição do equivalente computacional do sistema nervoso biológico, as redes neurais artificiais (RNA's). Para isto, apresentam-se as principais topologias de redes neurais e os tipos de comportamentos que cada topologia é capaz de gerar.

1.2 Objetivos

1.2.1 Objetivo Principal

O objetivo principal deste trabalho está relacionado com o desenvolvimento de agentes autônomos com alto grau de inteligência e autonomia. Pretende-se obter isto através do estudo, do desenvolvimento e utilização de redes neurais hierárquicas capazes de gerar e controlar os comportamentos de agentes autônomos.

A proposta, então, será o desenvolvimento de uma arquitetura de redes neurais hierárquicas onde as redes neurais diretas são responsáveis pela implementação do repertório de comportamentos básicos do AA, enquanto as redes recorrentes fazem a tarefa de seleção e coordenação dos comportamentos básicos, fazendo emergir um comportamento mais complexo e inteligente. Sendo a arquitetura hierárquica um estilo piramidal, as redes diretas serão implementadas na base da pirâmide (nível mais baixo) e as redes recorrentes implementadas em um nível acima.

1.2.2 Objetivos Secundários

Para alcançar o objetivo principal, necessita-se atingir os seguintes objetivos:

- O estudo aprofundado dos tipos de comportamentos, como os reflexivos e reativos, dos seres vivos;
- Estudo de como emergem os comportamentos, assim como, o entendimento dos passos entre o estímulo e a resposta, o comportamento;
- Verificar a existência de uma hierarquia no SNC, observando como é determinada a hierarquia dos comportamentos nos seres vivos;
- Uma vez verificada e entendida esta hierarquia, mostrar que ela é determinada por uma modularização, isto é, dá-se através de sistemas modulares independentes, mas interconectados.
- O estudo de algumas variações de implementações e métodos de treinamentos, como apresentado em ANDERSEN & MARTINEZ (1995, 1996ab), que servirão de base para ajudar na definição da nossa proposta.

1.3 Organização do Texto

O trabalho está organizado em 6 (seis) capítulos e 1 (um) anexo. No capítulo 1, a Introdução, descreve-se em um contexto geral, a motivação para o estudo e desenvolvimento, assim como, os objetivos da dissertação.

No segundo capítulo aborda-se os tópicos, redes neurais biológicas e noções de etologia, no qual procura-se fazer uma breve descrição da rede neural biológica e demonstrar que existe uma hierarquia no sistema nervoso central e que isto leva a comportamentos mais “inteligentes”.

Também, procura-se dar uma explanação sobre redes neurais modulares e, por fim, dar ênfase na hierarquia de comportamentos.

Existem várias alternativas de implementação para sistemas móveis. No terceiro

capítulo procuramos investigar mais aprofundadamente duas, por nos parecerem intimamente relacionadas com as alternativas aqui propostas. São elas: "Arquitetura de Subsunção" (Subsumption Architecture) de Brooks, e as "Criaturas de Braitenberg", proposta por Braitenberg.

No capítulo seguinte, capítulo 4, são abordados os controles hierárquicos dos comportamentos do robô, divididos em subtópicos. O primeiro é a rede hierárquica, capítulo 4, primeira rede implementada, com a descrição dos módulos e a prioridade dos comportamentos.

Em seguida, subtópico descrição dos comportamentos, procura-se mostrar que um repertório de comportamentos reflexivos (sistema sensorio e sistema motor) pode ser implementado através de redes neurais artificiais diretas e que, comportamentos reativos, coordenadores do repertório de comportamentos reflexivos, devem ser implementados através de redes neurais recorrentes. Neste trabalho, o aprendizado destas redes será em "offline", com o algoritmo de aprendizagem "Retropropagação" para treiná-las.

No subtópico alterações na estrutura da hierarquia, apresenta-se a nova hierarquia com o que foi modificado em relação a hierarquia anterior. A descrição dos novos comportamentos implementados, o próximo subtópico, não deixa de ser importante, pois, todas as expectativas em relação à funcionalidade das redes implementadas são descritas. O subtópico seguinte diz respeito a hierarquia dos autômatos de estado finito, os quais são representados pelas redes implementadas.

As definições e o aprendizado da primeira rede hierárquica implementada é apresentado no primeiro e no segundo subtópico do capítulo 5, trazendo as descrições de cada rede e dos seus conjuntos de treinamento. A análise dos resultados da primeira rede hierárquica implementada é o subtópico seguinte. Nele debate-se os resultados alcançados pela rede.

O subtópico seguinte, do capítulo 5, diz respeito ao treinamento e a definição de cada rede implementada, assim como a apresentação dos conjuntos de treinamento da

segunda rede hierárquica. O penúltimo subtópicos do capítulo 5 diz respeito aos novos experimentos e o que ocorreu com o robô quando foi colocado para se movimentar pelos ambientes. Por fim, no subtópico análise dos resultados, discorre-se sobre os resultados alcançados pela nova implementação.

No último capítulo, apresentam-se as conclusões, onde se procura dar ênfase sobre as dificuldades encontradas, o grau de satisfação alcançado com o trabalho, assim como, sugestões para novas implementações (trabalhos futuros) que podem ajudar no controle de agentes autônomos, tornando-os cada vez mais “inteligentes e autônomos”.

No anexo A, apresenta-se o simulador do robô Khepera, onde procuramos mostrar todo o funcionamento do ambiente, desde sua instalação até a implementação de sistemas que ele não dispõem e também de modificações que podem ser implementadas nos códigos que fazem parte do ambiente.

2 A EVOLUÇÃO BIOLÓGICA, ANATOMIA E FISIOLOGIA DO SISTEMA NERVOSO E COMPORTAMENTO ANIMAL

2.1 Evolução

“Em 1923, na Rússia, A. I. Oparin propôs que a atmosfera da Terra deveria ter sido, no passado, diferente da atmosfera hoje. Em particular, ela não deveria conter oxigênio, mas sim hidrogênio e outros compostos redutores, como o metano e a amônia. Oparin propôs que a matéria orgânica, de que depende a vida, havia-se formado espontaneamente numa atmosfera como essa, sob a influência da luz do sol, dos relâmpagos e das altas temperaturas existentes nos vulcões. Uma sugestão semelhante foi feita, independentemente, por outro materialista, J. B. S. Haldane, na Inglaterra. Estas duas propostas influenciaram a maioria dos autores, cujas especulações sobre as origens da vida foram publicadas nos anos trinta e quarenta. É surpreendente que a confirmação experimental demorasse tanto tempo, somente em 1953, que Stanley Miller, trabalhando com Harold Urey em Chicago, demonstrou que quantidades surpreendentemente grandes de compostos químicos se formavam, quando uma descarga elétrica passava através de uma atmosfera do tipo proposto por Oparin e Haldane”. (ORGEL, 1973)

Acredita-se, então, que através dos processos, descritos acima, apareceram as primeiras estruturas moleculares que tinham a capacidade de auto-replicação. Esta replicação dava-se a partir da catalisação de materiais de outras moléculas e elementos químicos, retirados do ambiente. Esta hipótese deu o Prêmio Nobel de Química (um prêmio político, devido ao início da guerra fria) a Harold Urey, pois seu aluno S. Miller fez circular uma mistura de vapor de água, metano, amônia e hidrogênio, gases (atmosfera proposta por Oparin), sob intensas fagulhas elétricas. Após uma semana constatou o aparecimento de glicina e alanina, dois dos aminoácidos constitutivos das proteínas. (WALD, 1973).

Pode-se imaginar que com o tempo, depois da glicina e alanina, outros compostos orgânicos, apareceram até que apareceu uma molécula com autossimetria, ou seja, capaz de se auto-replicar. Para isto necessitava-se de moléculas mais simples e iniciou-se o que costumamos chamar competição: as moléculas auto-replicáveis, que tinham formas que facilitassem a captação de moléculas primitivas, tinham mais probabilidade de se reproduzir vencendo a “competição”.

Como resultado da competição entre as moléculas veio o aparecimento da célula. Entre outras características, a célula era capaz de se proteger, estendendo a região de influência de uma molécula auto-replicável até uma membrana externa e de executar processos metabólicos com elementos tirados do ambiente, usando suas complexas estruturas internas. Segundo CHURCHLAND (1984), o conceito padrão de vida passou a existir com a emergência da célula, um sistema consumidor de energia que se automantém e se autoduplica.

Pode-se dizer que a Evolução, baseada na Seleção Natural, foi o mecanismo que a Natureza encontrou para aumentar a complexidade dos seres, tornando-os capazes de sobreviver em ambientes adversos, pois sendo o ambiente finito, existe uma competição entre aqueles que o habitam. A Natureza, através da evolução, obteve uma diversidade de organismos biológicos, e estes indivíduos interagem com o meio ambiente.

“O Neo-Darwinianismo defende a idéia de que a história de todas as criaturas vivas pode ser explicada através de alguns processos estatísticos atuando sobre populações e espécies. Estes processos são: reprodução, mutação, competição e seleção.” (ROISENBERG, 1998)

“Supondo-se válida a teoria da evolução biológica, deve-se notar que este processo não é um processo movido por uma finalidade exterior ao processo de evolução em si. Em outras palavras, não existe evidência de que a finalidade da evolução seja produzir a humanidade ou outra espécie qualquer. A evolução é o resultado da diversidade de organismos biológicos que interagem com o meio ambiente, retirando deste meio as suas condições de vida e, por sua vez, modificando-o”.(BARRETO, 2000)

Neste ambiente, os adaptados, aqueles que mais se capacitam, encontrando condições de vida e maiores possibilidades de reprodução, tendem a aumentar o número de sua espécie, dominando o ambiente. A isto damos no nome de *Seleção Natural*.

A seleção levou à evolução das espécies, causando o aparecimento de sistemas com estruturas progressivamente mais complexas. Conforme houve o aumento da complexidade do sistema, houve também o aumento da complexidade de seus comportamentos. Para controlar estes comportamentos surgiu um mecanismo, o Sistema Nervoso, que juntamente com os músculos formam um sistema inteligente de seleção e coordenação dos comportamentos dos seres vivos. KOVÁCS (1997) descreve de forma clara esta junção, além de outros detalhes do SNC.

A origem e as etapas do processo de desenvolvimento e especialização deste mecanismo não são totalmente conhecidas. No seu livro – “Matter and Consciousness” - CHURCHLAND (1984) apresenta uma situação hipotética que exprime, com bastante didática, estas etapas.

“Para apreciar quão facilmente um sistema nervoso pode vir a determinar aptidões de sobrevivência de toda uma espécie, consideremos uma criatura parecida com um caramujo que habitasse o fundo dos oceanos primitivos, há muitos milhões de anos. Esta espécie deveria sair da sua concha para se alimentar e recolher-se a ela apenas para descansar ou quando algum corpo externo fizesse contato direto com ela, como por exemplo, quando um predador atacasse. Muitas destas criaturas eram devoradas por predadores, pois, apesar do comportamento reflexo de se recolher à sua concha, muitas eram mortas no primeiro ataque. Deste modo, as populações de diferentes espécies eram estáveis, estando em equilíbrio com a população de predadores.

Enquanto isto acontecia, todos os caramujos desta espécie possuíam uma faixa com células sensíveis à luz no topo de suas cabeças. Não existe aí nenhum fato notável. Muitos tipos de células são sensíveis à luz em diferentes graus. A presença destas células era uma característica completamente indiferente à espécie, pois não tinham nenhuma função.

Suponhamos agora que, devido a uma mutação no código genético de um determinado caramujo, células nervosas, ligando a superfície da pele aos músculos que o faziam recolher-se à sua concha, tenham crescido exageradamente, eventualmente conectando as células sensíveis à luz com estes músculos. Assim mudanças bruscas na iluminação fariam com que o caramujo se recolhesse imediatamente a sua concha.

Esta nova característica deste caramujo poderia não ter nenhuma importância em vários ambientes. No entanto, nas condições ambientais existentes para aqueles caramujos, mudanças bruscas de iluminação normalmente eram causadas por predadores nadando diretamente acima das suas cabeças. Nosso caramujo mutante possui agora um sistema de alerta antecipado, que permite que ele se recolha à sua concha quando um predador se aproxima para atacá-lo. Suas chances de sobrevivência e conseqüentemente de se reproduzir são muito maiores do que a dos companheiros não atingidos pela mutação. Uma vez que esta alteração é fruto de uma alteração genética, muitos dos seus descendentes herdarão esta característica e, portanto, também serão maiores suas chances de sobreviver e se reproduzir. Claramente, com o tempo, esta característica passará a dominar a população de caramujos. De pequenos e fortuitos eventos é que são feitas as grandes mudanças.

Evoluções futuras podem ser facilmente previstas. Se por mutações genéticas a superfície sensível à luz se curvar sobre uma cavidade hemisférica, suas porções seletivamente iluminadas fornecerão informação direcional a respeito das fontes e oclusões de luz; informações estas que podem acionar respostas motoras direcionais. Em uma criatura móvel, como o peixe, por exemplo, esta característica traz ainda mais vantagens, seja como caça, seja como caçador. Num próximo passo, a cavidade hemisférica pode se transformar em uma cavidade esférica com as células sensíveis à luz no fundo, e um pequeno buraco para o mundo exterior, permitindo que as imagens então se formem na superfície sensível. Eis então que 'miraculosamente' um olho foi criado." (CHURCHLAND, 1984).

2.2 Anatomia e Fisiologia do Sistema Nervoso

A proposta da seção 2.2 é mostrar a parte anatômica e fisiológica dos sistemas nervosos, buscando principalmente, mostrar a existência de uma hierarquia na organização dos sistemas nervosos.

Experimentos feitos por alguns pesquisadores no século XIX, com cães, e citados nesta seção, buscaram decifrar alguns comportamentos e como eles eram gerados pelo sistema nervoso.

Encontrar uma hierarquia no sistema nervoso não era o objetivo dos experimentos, porém, os resultados alcançados mostram que o cérebro não comanda alguns comportamentos e que os comportamentos de arco reflexo, na sua maioria, são controlados pela medula espinhal. A existência de diferentes níveis para a geração de comportamentos diferentes ficou evidente.

Mesmo que o sistema nervoso seja uma grande cadeia única, sem separações, o que podemos perceber é que existem módulos que processam alguns tipos de informações e também existem vários módulos para processar apenas um tipo de informação.

Os módulos podem ter sua autonomia para atuarem sobre os sistemas que estão sobre sua regência, contudo, quem comanda qual módulo ou qual comportamento emergirá é o “SNC” que seleciona e coordena os comportamentos.

2.2.1 Anatomia

Sabe-se hoje que os neurônios e as sinapses, base do sistema nervoso, estão completamente desenvolvidos em sistemas simples como celenterados (hidras e anêmonas-do-mar), em sistemas um pouco mais complexos como, equinodermos (estrelas-do-mar e ouriços-do-mar), nos sistemas nervosos bilaterais (vermes e moluscos), assim como nos vertebrados.

Como citado acima, podemos encontrar os neurônios em formas multicelulares primitivas, é o caso das hidras. Nestas formas o sistema nervoso consiste de uma fina cadeia de células nervosas individuais não apresentando cérebro ou qualquer outra forma especial que faça o papel de um mecanismo coordenador.

Na medida em que os organismos se tornam mais complexos, aumenta a especialização das células nervosas. Com isto aparecem algumas que conduzem os estímulos da superfície para a região central do organismo e são chamadas de sensitivas ou aferentes.

Nesta região central elas entram em contato com outro neurônio chamado de motor ou eferente que, por sua vez, faz a condução do impulso da região central até as células efectoras, produzindo a ação desejada.

Para coordenar as atividades de acordo com os estímulos recebidos através dos neurônios sensitivos, foi desenvolvido um mecanismo integrador chamado de Sistema Nervoso Central (SNC). Esta coordenação é produzida por neurônios interpostos entre os neurônios aferentes e eferentes, os quais se denominam Interneurônios ou Neurônios Conectivos.

Em grandes animais, a maior parte do SNC (cérebro e medula espinhal) é composta por estes interneurônios, onde estão conectados neurônios de vários níveis ou segmentos, que integram suas atividades. (GETTY, 1986), (GANONG, 1989), (FRANDSON, 1979), (WOOD, 1973), (SCHMIDT-NIELSEN, 1976,1996).

Então, a primeira consideração a ser levada em conta é: o sistema nervoso faz a coordenação de todas as atividades do organismo. Isto significa que ele integra as sensações e “idéias”, ajusta “fenômenos de consciência” e adapta o organismo às condições do momento (ERHART, 1974).

Podemos dividir o sistema nervoso em:

- (1) Central, inclui o cérebro e a medula espinhal;

- (2) O sistema nervoso periférico, consistindo em nervos cranianos e nervos espinhais, em direção às estruturas somáticas (corpo), e o sistema nervoso autônomo que segue em direção às estruturas viscerais.

Devemos ressaltar que as divisões relacionadas ao sistema nervoso, principalmente dos mamíferos, não passam de uma convenção para uma melhor descrição deste sistema, pois ele é uma unidade integrada.

“O sistema nervoso é composto pelo cérebro, medula espinhal e nervos periféricos que se estendem através do corpo. A função principal do sistema nervoso é de controlar muitas atividades corporais, sobretudo os músculos; mas, para manter esse controle com inteligência, o cérebro necessita ser permanentemente notificado de tudo aquilo que rodeia o organismo” (GUYTON, 1974).

Esta definição é relacionada aos Homens, mas não exclusiva, como mostra a fig.

2.1.

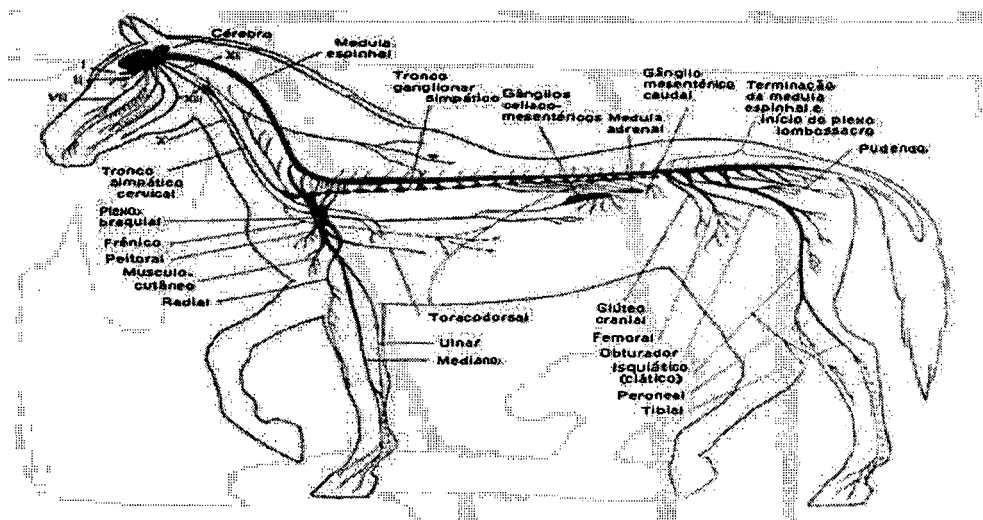


Figura 2.1: Sistema Nervoso do Cavalo

Para executar a função descrita por GUYTON (1974), o sistema nervoso depende de duas partes separadas, a parte sensorial e a parte motora, pois qualquer evento que venha causar uma perturbação ao redor do organismo é registrado pelos órgãos de sentido (parte sensorial). Este registro é feito através dos sentidos da visão, audição,

olfato, paladar e tato. Mas faz-se necessário o aprendizado para o ser vivo aprender a usar esta parte sensorial, isto é, não adianta sentir o cheiro ou visualizar o perigo se seu sistema não está apto a reagir a isto. Em SAKSIDA (1999) encontramos algumas comparações de aprendizados.

Na parte motora temos os músculos que, ao contrário do que pensamos, não são totalmente controlados pelo cérebro. Grande parte das atividades musculares é controlada pela medula espinhal e parte inferior do tronco cerebral, que atuam em nível subconsciente.

Olhando a organização do sistema nervoso, vemos que cada neurônio consiste em um pericário ou soma (corpo da célula nervosa) e em um ou mais processos nervosos, chamados de dendritos e axônios, como mostra a fig. 2.2. Toda esta organização é regida por estas células especializadas, as quais são consideradas unidades anatômicas e funcionais, citadas acima como sendo a base do sistema nervoso, chamadas de Neurônios ou Neurócitos.

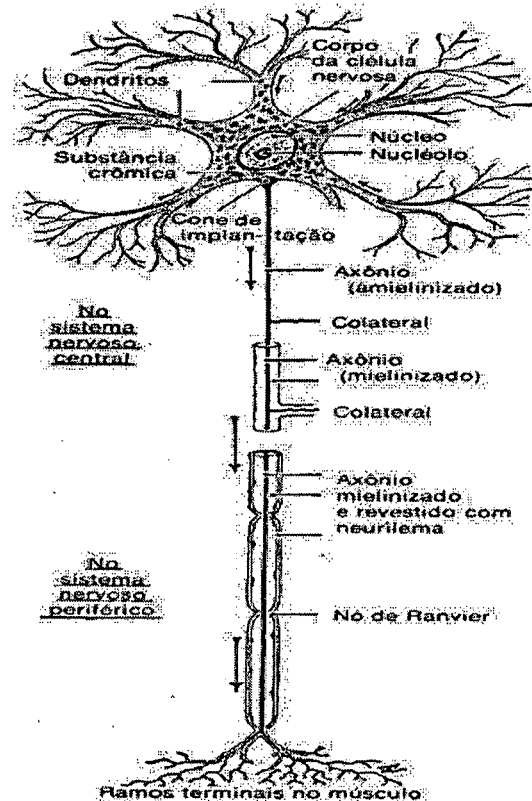


Figura 2.2: Estrutura de um Neurônio.

"Cada neurônio consiste em um corpo celular (isto é, Pericário) do qual um ou mais prolongamentos se estendem por distâncias variadas. A característica mais notável do neurônio é a existência de prolongamentos protoplasmáticos, alguns dos quais se estendendo por longas distâncias. O neurônio é definido como uma célula nervosa, assim como todos os prolongamentos que dela se originam" (CARPENTER, 1976).

Estes neurônios podem ser classificados em unipolar, bipolar e multipolar, devido ao número de seus prolongamentos (axônios e dendritos). Geralmente o núcleo celular é esférico e envolvido pela membrana nuclear, com sua posição predominantemente no centro. Alguns são binucleados.

Os prolongamentos:

- Dendritos - expansões protoplasmáticas do pericário, muito ramificados, curtos, com superfície às vezes rugosa.
- Axônios - são lisos e originam-se do pericário em um pequeno cone (chamado de implantação ou cone axonal) tornando-se comumente longo e delgado. Podem ser mielinizados ou amielínicos; o primeiro possui uma bainha de mielina em parte do seu trajeto ou em todo ele. Esta bainha tem sua constituição a pouca distância do corpo celular e sua espessura é variável. Ela é composta de camadas concêntricas onde se alternam moléculas proteína e lipídio.

Os neurônios fazem conexões através de seus axônios ligados aos dendritos de outros, e a esta ligação damos o nome de **sinapse** (CONTU & OSÓRIO, 1972). Esta ligação pode ser feita através de qualquer parte do neurônio, mas na sua grande maioria o neurônio pré-sináptico forma intumescências terminais ou *nó sináptico* nos neurônios pós-sinápticos. Há casos em que um neurônio tem seu soma coberto, em até 80% da sua extensão, por nós sinápticos produzidos pelos dendritos de outros neurônios. (GETTY, 1986), (GANONG, 1989), (FRANDSON, 1979), (WOOD, 1973), (SCHMIDT-NIELSEN, 1976,1996).

2.2.2 Fisiologia

Podemos destacar duas propriedades, a irritabilidade e a condutividade, associadas à vida dos organismos unicelulares, tal como a ameba. Estas propriedades dependem da receptividade desta célula única, isto significa que ela deve ser receptiva a estímulos, como: alterações de luminosidade, temperatura, pressão osmótica, pH e pressão.

Ser receptiva a estes estímulos significa ser irritável. Aliado à irritabilidade tem-se a propriedade ou capacidade de condutividade, a qual ocorre quando os estímulos são recebidos e transmitidos para a porção da célula que executará a ação apropriada. Temos nesta transmissão a representação da condutividade.

Estudando a evolução dos seres, pode-se verificar que organismos simples evoluíram para formas mais complexas e, assim como ocorreu esta evolução dos organismos, vemos que algumas células especializadas, precursoras das células nervosas (os neurônios), desempenham as duas atividades descritas acima.

Então, chegar no que hoje conhecemos como neurônio, que é irritável e condutivo, também dependeu de uma evolução e especialização de algumas células.

Para o entendimento do funcionamento do sistema nervoso central, nota-se a necessidade de discorrer sobre a fisiologia destas células. Contudo, o assunto é bastante extenso e ter-se-ia que delinear suas propriedades que incluem absorção, excreção, secreção, a própria irritabilidade, condutividade e contratilidade. Estas duas últimas, características das células nervosas e musculares, respectivamente.

Portanto, pela extensão do assunto, cabe aqui apenas a indicação de literaturas que discorrem sobre estas propriedades, como encontrado em GANONG (1989), SCHMIDT-NIELSEN (1976) e WOOD (1973), uma vez que o intuito deste trabalho não é mostrar os processos internos do sistema nervoso, como velocidade dos impulsos ou os processos químicos que ocorrem, e sim, o controle que ele exerce sobre o corpo animal.

A importância da coordenação nervosa para nós supera os processos internos do sistema nervoso, pois os processos químicos são os mesmos em um neurônio de um

celenterado e de um mamífero, mas a coordenação depende do que ocorre ao seu redor, apesar das reações serem muito parecidas quando olhamos o arco reflexo.

Assim como filósofos da antiguidade e da idade média procuravam, de certa forma sendo insensatos, um local onde a alma estivesse abrigada e a função no encéfalo, hoje procuramos entender como o sistema nervoso faz o controle das ações.

Com a ajuda da tecnologia tornou-se possível a visualização das células, neurônios, e do seu funcionamento. Assim sendo, experiências feitas com cães no século XIX, época na qual surgiu o conceito de reflexo, também pode ter dado início ao conceito de hierarquia, pois estas consistiam em isolar o tronco e as pernas através de uma secção entre o encéfalo e a medula.

O significado disto foi que, após ser feito o corte, aplicou-se um estímulo em uma das patas e houve uma resposta, isto é, o estímulo provocou uma dor e a consequência foi uma contração dos músculos.

Esta experiência comprova que grande parte dos arcos reflexos e outras funções não são controladas pelo cérebro, e sim pela medula. Outras experiências e estudos, os quais serão comentados a seguir, também desempenharam um papel importante para o entendimento do SNC.

Algumas das experiências foram feitas usando animais primitivos, nos quais o controle de suas ações é realizado por uma rede nervosa, como no caso citado da Hidra, que une órgãos de sentido e musculatura contrátil.

Além de ser capaz de fazer alguma coordenação, ela é capaz de ter movimentos que levam à captura e à ingestão de pequenos crustáceos.

Tomando agora uma anêmona-do-mar para observação, estimulando um de seus tentáculos, com uma agulha, ele curva-se em direção à boca. Se continuarmos estimulando, pode-se perceber que os tentáculos vizinhos acompanham o movimento.

Uma resposta esperada, pois através de uma condução chamada de decremental, vemos o estímulo sendo disseminado lentamente a partir do ponto de sua aplicação.

Contudo, não é o único tipo de resposta que devemos esperar de uma anêmona, já que estimulando, com um bastonete de vidro, o disco pedal, veremos que ela irá se fechar rápida e completamente.

Com isto, é perceptível que houve uma resposta síncrona e simétrica em relação ao eixo do corpo.

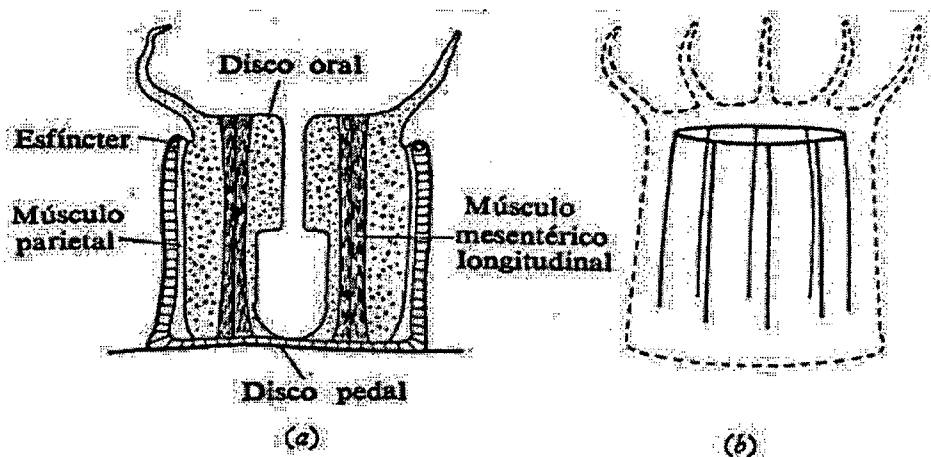


Figura 2.3: Estrutura do sistema nervoso da Anêmona-do-mar.

“Nos equinodermos encontramos indícios de centralização nervosa, no sentido anatômico, e o aparecimento de algumas das propriedades fisiológicas associadas ao S.N.C. Na estrela-do-mar, o S.N.C. tem a forma de cordões nervosos situados no sulco da superfície ventral de cada braço e de um anel em redor da boca. Estes cordões são um pouco mais que condensações da rede nervosa, porém em associação próxima com ele estão as células nervosas motoras que inervam os músculos. A organização do sistema, como sistema nervoso central, é bastante fraca; apesar disto, a coordenação dos movimentos do animal depende de sua integridade”. (RAMSAY, 1973).

No caso das estrelas-do-mar, sua locomoção ocorre através de centenas de pés ambulacrários situados e distribuídos na parte inferior dos braços. O braço líder nem sempre é o mesmo e como consequência disto, tem-se então, o conceito de inibição no sistema nervoso da estrela-do-mar. Se pressionarmos o braço líder, a reação será a de cessar os passos para aquela direção e recomeçar em uma outra, havendo uma modificação na liderança, ou seja, outro braço assume a liderança. Os pés ambulacrários

do braço líder seguem um movimento longitudinal em relação ao braço, e os pés dos demais braços seguem um movimento paralelo ao braço líder.

Efetuada dois cortes de maneira que isolem dois braços dos demais, os pés ambulacrários, nestes dois, darão passos paralelos e os pés dos outros três também darão passos paralelos, mas não significa que será para a mesma direção.

Os sistemas dos celenterados e dos equinodermos, descritos até este momento, são muito simples e por isto veremos alguns outros na seqüência, que podem acrescentar um pouco mais ao nosso conhecimento nesta área. O primeiro a ser visto será o sistema dos anelídeos. O SNC dos anelídeos é composto pelo cérebro e dois cordões nervosos ventrais, normalmente unidos com gânglios em cada segmento, e os nervos que saem dele são curtos, porém suficiente para conduzir corrente de ação.

O representante dos anelídeos a ser descrito aqui é a minhoca. Sua locomoção é semelhante a um intestino em peristalse, porém ela é feita com alguma coordenação.

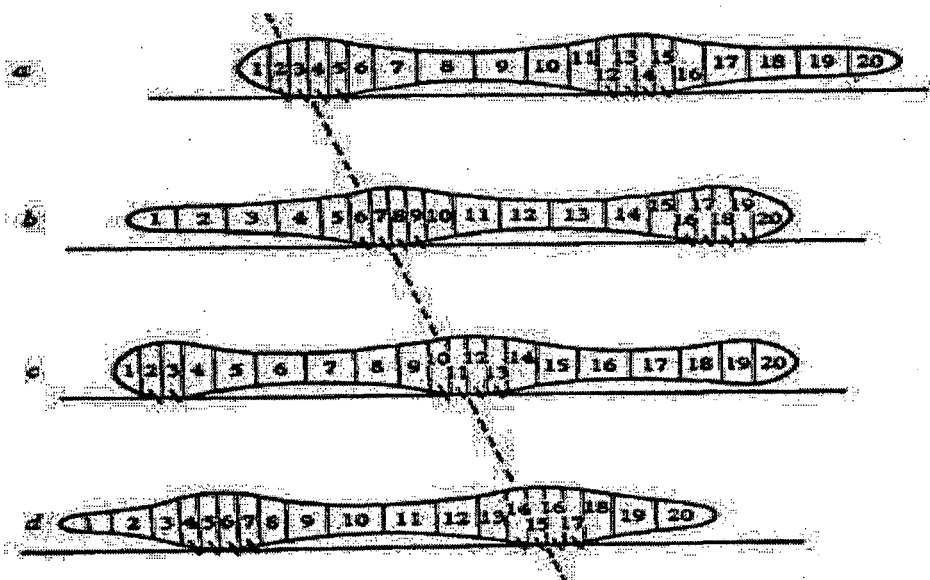


Figura 2.4: Locomoção da minhoca

“A cavidade do corpo da minhoca é dividido transversalmente por septos. Cada septo é uma folha muscular que separa completamente um segmento do seguinte, exceto um pequeno orifício na parte ventral. Este orifício é circundado por uma concentração muscular, semelhante a um esfíncter, em

geral totalmente contraída, sendo o septo, para todos os efeitos e propósitos, completo. Se os septos fossem anteparos rígidos, à prova d'água, a pressão existente no celoma de qualquer segmento seria determinada apenas pela musculatura do próprio segmento. A pressão exercida pelos músculos, em um segmento, não poderia ser transmitida ao longo do corpo e ser usada para distender músculos em outra parte. A presença dos septos oferece certo grau de independência aos segmentos individuais". (RAMSAY, 1973).

Dirigindo nossas atenções para o SNC da minhoca e para as conexões das fibras nervosas associadas a ele, vemos que, além das fibras gigantes, existem as fibras nervosas que são curtas em relação ao corpo deste animal.

"Os corpos celulares dos neurônios motores estão agrupados nos gânglios segmentares e seus axônios inervam as fibras musculares do mesmo segmento. Os corpos celulares dos neurônios sensitivos situam-se na periferia, perto dos órgãos dos sentidos, e seus axônios penetram no SNC, terminando geralmente no gânglio do mesmo segmento. Além disso, há muitos neurônios internunciais, cujos axônios estão totalmente confinados ao SNC, percorrendo uma curta distância ao longo do cordão, de um gânglio para outro". (RAMSAY, 1973).

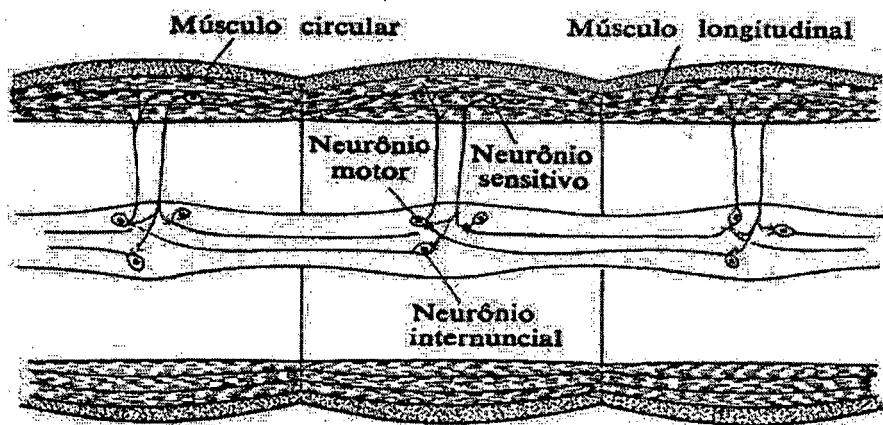


Figura 2.5: Estrutura das conexões nervosas em uma minhoca.

Na minhoca encontramos um sistema direto de condução de impulsos, e isto é feito por três fibras gigantes situadas na parte dorsal do cordão nervoso. Enquanto a

fibra central tem seu diâmetro em cerca de 50 μ e velocidade de impulsos em cerca de 30 m/s, as fibras laterais tem seu diâmetro em cerca de 25 μ e velocidade de condução em 10 m/s. Estas fibras são formadas pelos axônios de muitas células nervosas, dispostas ao longo do cordão. Cada segmento dos músculos longitudinais está ligado às fibras gigantes através de fibras nervosas, sendo estas fibras nervosas originadas nas fibras gigantes.

As fibras laterais estão ligadas entre si e se um impulso é originado em uma delas, este é propagado para a outra. A fibra mediana está fisiologicamente separada, mostrando que existem dois sistemas separados de condução direta. O propósito destes sistemas é produzir conexões diferentes com outras partes do corpo da minhoca, assim, se aplicarmos um estímulo em qualquer parte de um dos sistemas, irá provocar uma contração de todos os músculos longitudinais. Qual seria a vantagem disto? O fato dos sistemas fazerem conexões diferentes em outras partes do corpo proporciona reações diferentes, isto é, se aplicarmos um estímulo na extremidade cefálica, provocará impulsos na fibra gigante mediana, conseqüentemente, um estímulo aplicado na extremidade caudal, provocará um impulso nas fibras laterais.

Qual é o sentido desta inversão? Imagine que um estímulo doloroso seja aplicado na extremidade cefálica. A estimulação da fibra gigante mediana causa um protraimento das cerdas na extremidade cefálica e a estimulação nas fibras laterais provoca o protaimento nas cerdas da parte caudal. Este estímulo provocaria uma retração na parte cefálica, porém isto depende do suporte feito pela cauda. Em outras palavras, enquanto um estímulo na parte cefálica provoca uma retração desta parte, provoca também uma contração na cauda, a qual é “fixada” ao “chão” para fazer suporte à parte cefálica. O contrário também ocorre, isto é, se o estímulo ocorrer na parte caudal o processo é invertido.

Comparando os sistemas nervosos da anêmona e da minhoca, vemos que os dois animais têm similaridades nos seus mecanismos, uma vez que existe a condução lenta de impulsos, célula a célula, e existe o sistema de condução rápida que entra em ação quando o animal se vê ameaçado. Encontramos aqui dois sistemas diferentes de condução direta, em dois tipos de sistemas nervosos diferentes, mas como o mesmo

propósito: autopreservação.

Encontramos as fibras gigantes também nos artrópodes, aqui representados pela barata. Neste animal as fibras gigantes estão situadas no cordão nervoso que vão do abdome até os gânglios torácicos. A barata é capaz de sentir o deslocamento do ar e isto torna-se possível pelo fato das cerdas dos cercos anais, responsáveis pela detecção do deslocamento, estarem em conexão funcional com as fibras. O resultado desta função na barata é sua rapidez para fugir do “perigo”.

Se tentarmos matar uma barata e não formos rápidos para isto, ela fugirá. No momento em que seus cercis sentem o deslocamento do ar, provocado pelo nosso movimento, um impulso é gerado atingindo as fibras gigantes. A função das fibras é fazer com que a barata corra com velocidade máxima, fugindo do perigo e se autopreservando.

Este exemplo comprova a existência dos arcos reflexos de tônus plástico. Se for aplicada uma força à perna, com efeito semelhante ao de um peso colocado na parte dorsal, os órgãos dos sentidos aumentam sua frequência de descarga, podendo, se for usado um aparelho para medir frequência, ser captadas séries mais rápidas de impulsos através dos músculos apropriados que sustentam o corpo.

Estes animais citados são importantes para este trabalho, no entanto, a contribuição dos vertebrados é maior, pelo fato de apresentarem o sistema nervoso, a organização modular e hierarquia, mais complexo. Nos vertebrados o sistema nervoso é composto pelo cérebro, medula e nervos, os quais ligam as partes centrais com todas as áreas do corpo.

“Geralmente, referimo-nos ao cérebro e à medula como Sistema Nervoso Central. Esse sistema recebe informação dos vários órgãos dos sentidos através de nervos denominados nervos sensitivos. O sistema nervoso central também envia impulso às várias partes do corpo. Alguns nervos estimulam os músculos usados para a movimentação e por isso são denominados nervos motores. Outros nervos extremamente importantes, se bem que não estejam sob a ação do controle voluntário, pertencem ao sistema nervoso

autônomo que regula as funções dos órgãos internos, tais como as glândulas digestivas, o coração, o intestino, etc". (SCHMIDT-NIELSEN, 1976).

Nos animais mais evoluídos encontramos o que chamamos de *Centros Superiores*. A partir de experimentos efetuados em animais e observações feitas em humanos com lesões cerebrais, foi possível perceber que o córtex, parte mais externa do cérebro, é necessário para a percepção consciente, associação e pensamento. E que a audição, visão, fala, movimentações de algumas partes do corpo, etc, são associadas a partes específicas do córtex.

Saindo do cérebro e estendendo-se na cabeça temos os chamados *Nervos Cranianos* que formam doze pares de nervos. Os *nervos medulares*, que são um conjunto de nervos emitidos pela medula a partir da sétima vértebra, suprem todas as porções inferiores do organismo. Por exemplo, se sentimos dor em um dos dedos da mão, esta sensação é transmitida primeiramente ao nível da sétima vértebra do pescoço, mas se movemos a mão em um simples ato de abano, o comando sai do cérebro e passa pela medula e através dos nervos motores para ser executado. Com isso vemos que apesar da grande autonomia da medula, ela está sob o controle do cérebro.

Os nervos sensitivos responsáveis pela condução dos impulsos até o sistema nervoso são divididos em: nervo olfativo (primeiro craniano) - responsável pelo sentido do olfato; nervo óptico (segundo craniano) - conectado à retina; nervo trigêmeo (quinto craniano) - responsável pelos impulsos advindos da testa, couro cabeludo, pálpebra, nariz e dos dentes, cada fibra conduz um tipo de informação para o cérebro, e isso caracteriza a presença de fibras motoras; o nervo facial (sétimo par) - este também é um nervo misto, pois conduz o sentido do gosto.

O nervo acústico (oitavo nervo) - responsável pelo sentido da audição e do equilíbrio, sendo este último relacionado com as estruturas internas do ouvido; o glossofaringeo (nono craniano) - também misto, tendo a função de conduzir os sentidos do paladar e do tacto; o último deles, nervo vago (décimo craniano), mas extremamente importante, uma vez que faz a conexão com a maioria dos órgãos internos. Com

exceção da dor, pouca informação vinda dos órgãos internos é tornada consciente.

No caso dos nervos motores, podemos dividir em dois grupos: os nervos que estão sobre o controle voluntário, os quais suprem os músculos estriados ou esqueléticos e os nervos que não estão sobre o controle voluntário, os quais levam todos os outros impulsos eferentes. Na verdade, todos os músculos podem ser controlados conscientemente, mas a maior parte de suas atividades nunca atingem o plano da percepção consciente. Podemos tomar como exemplo os complexos movimentos da respiração e do andar que ocorrem sem pensarmos neles. Um outro sistema encontrado é o sistema nervoso autônomo, que é parte do sistema nervoso periférico e não está sujeito ao controle da vontade.

O certo de tudo que falamos é que o corpo é controlado pelo Sistema Nervoso Central (SNC). Ele é quem impulsiona os músculos. A importância da saída motora final (comportamento), reside no fato de ser adaptativa, como descrito em MESSENGER (1980). O sistema nervoso determina quais são os músculos que serão ativados.

Um tipo de controle considerado importante é o de retroação (malha fechada), onde a saída do sistema é usada para influenciar a entrada. Os biólogos costumam usar os termos *Retroação e Feedback*. O que basicamente acontece é o controlador (SNC) que tem como prioridade, em termos gerais, as instruções “sobreviva e procrie”, ele recebe as informações dos órgãos dos sentidos, processa esta informação, ajustando a saída do “sistema controlado”, fazendo com que sejam neutralizadas as situações que poderiam ser prejudiciais ao animal.

Mas quando se fala de animais considerados superiores isto fica mais complexo, pois o SNC dos Humanos é mais complexo, uma vez que nos oferece canais múltiplos, isto é, as informações são processadas em paralelo em diferentes áreas do nosso cérebro, onde devemos esperar encontrar sistemas distintos de controle para diferentes atividades. Um exemplo é que o controlador dos músculos da perna e o dos olhos são diferentes, mas não estão totalmente independentes.

Isto nos leva à conclusão que o circuito fechado para operações complexas não

pode ser muito simples. Assim sendo, precisa existir um sistema, em nível superior, que controle os subsistemas (sistemas de nível inferior), isto é, há necessidade de controlar o controlador de algumas atividades corporais. (MESSENGER, 1980).

2.3 Redes Neurais Modulares

A modularização está presente em vários organismos vivos. Para entendermos um pouco mais, basta olharmos para as células que junto com outras milhões formam “um corpo” ou até mesmo os órgãos de um animal que trabalham em sua função, porém, seu trabalho é para manter o corpo em ordem. Então podemos ver um órgão como sendo um módulo que, juntamente com outros órgãos, ajuda a formar o todo (um animal).

Este foi um exemplo, mas, todas as partículas elementares fazem, de certa forma, parte de outro sistema que elas ajudam a formar. Então, *“estas partículas são módulos de sistemas maiores que formam uma ordem progressiva, por exemplo; átomos, moléculas, sólidos e fluídos, corpos celestiais, sistemas estelares, galáxias, agrupamentos, superagrupamentos até o próprio universo”*. (BOERS & KUIPER, 1992)

Aqui será tratado especificamente de sistemas modulares existentes em animais, mais precisamente a modularização existente no sistema nervoso. Sabemos que os animais também são formados de partículas elementares, isto é, em todas as possíveis classes os organismos vivos podem ser formados de átomos, moléculas, fluídos e sólidos. Por exemplo, se dividirmos um animal até chegarmos nas células, veremos que elas também se dividem em organismos menores que também executam funções.

“Os mamíferos, por exemplo, podem ser subdivididos em suas células, cada uma das quais podem ser subdivididas em várias organelas (partes especializadas de uma célula) posicionadas no núcleo e no citoplasma (o primeiro corpo vivo de uma célula não incluindo o núcleo). Estas células são agrupadas para formar órgãos, os quais em troca formam o organismo inteiro. No organismo cada órgão tem uma ou mais funções específicas.

Cada órgão é dividido novamente em partes com funções diferentes, e nessas partes cada célula tem suas próprias tarefas. Estes exemplos sugerem uma modularização recursiva dentro da sua própria modularização” (BOERS & KUIPER, 1992).

Esta modularização pode ser dividida em duas partes distintas, repetitiva e diferenciada. A modularização repetitiva refere-se ao uso do mesmo tipo de módulo em tempos distintos e a modularização diferenciada ao uso de módulos distintos para a organização de um todo.

Nosso cérebro trabalha com esta modularização, pois todos os processos podem ser executados em paralelo ou não. Por exemplo, se alguém está escrevendo um livro, seja manuscrito ou digitado, ele trabalha com as mãos, mas as idéias estão sendo processadas pelo cérebro. Isto significa que conforme as idéias aparecem, o cérebro envia o comando para as mãos escreverem. São praticamente dois processos distintos que estão sendo executados em paralelo.

Imagine que além de estar escrevendo um livro, estamos escutando uma música. Parece algo fácil, mas os processos que estão em execução já não são dois. Agora estamos processando idéias, organizando-as, digitando-as e, ao mesmo tempo, estamos tendo a percepção musical e também o possível entendimento da mesma. Passando, então, à execução de vários processos simultâneos. Podemos também acrescentar outros processos que poderiam estar sendo executados durante todos estes anteriormente citados.

O estudo desta modularização leva a percepção de que existirão casos nos quais serão usados os mesmos módulos para executar processos diferentes em tempos distintos. E existirão casos em que um mesmo processo estará sendo executado por módulos diferentes, em diversas regiões do cérebro ao mesmo tempo.

Estes sistemas modulares, no cérebro, podem ser de dois tipos, vertical e horizontal. A estrutura horizontal é encontrada nos processos executados por camadas hierárquicas subseqüentes de neurônios. Este estágio pode, por exemplo, ser

identificado no sistema visual primário, onde simples características de imagens visuais como linhas e arcos são representados em camadas de neurônios simples. Estes são combinados e são representados através de neurônios em camadas subseqüentes.

A estrutura vertical se dá na existência, para todos os níveis do cérebro primata, de caminhos múltiplos de processos paralelos. Esta estrutura permite o processo separado de tipos diferentes de informação.

“Um bom exemplo pode, mais uma vez, ser achado no sistema visual, onde diferentes aspectos de incentivos visuais como forma, colorido, movimento e posição, são processados em paralelo através dos sistemas neurais anatomicamente separados, organizados no caminho magno celular e parvo celular. Estruturas convergentes integram estes processos de informação visual separadamente no mais alto nível hierárquico para produzir uma percepção unitária” (BOERS & KUIPER, 1992).

Um indício da existência da organização modular é a presença de ambas as estruturas, horizontal e vertical, no cérebro. Uma macro visualização de uma organização modular é a existência dos dois hemisférios no cérebro, pois sabe-se que existem grupos inteiros de funções mentais localizados em apenas um dos hemisférios. Existem funções sendo executadas em regiões diferentes do mesmo hemisfério. Análises das funções de comportamentos indicam que até as mais complexas funções do cérebro podem, de certa forma, serem localizadas. Isto leva a estudos que permitem a definição das regiões, podendo ajudar na tentativa da eliminação de alguns problemas ou afasias.

Um destes estudos revelou que danos cerebrais localizados foram resultados de lesões locais, isto é, alguns danos cerebrais provocavam a perda de algumas habilidades, mas outras não eram afetadas. Um exemplo é a afasia de Wernicke que causa a incapacidade no entendimento de idiomas escritos ou falados, enquanto os pacientes podiam perfeitamente falar e escrever, de forma ininteligível, mas fluentemente. A afasia afetou a parte do cérebro responsável pelo entendimento do idioma, mas, não afetou a parte da fala, por exemplo.

Uma vantagem funcional importante da separação anatômica das diferentes funções é a minimização da interferência mútua entre os processos simultâneos e execução de tarefas diferentes, necessárias para executar habilidades complexas como, por exemplo, dirigir um carro. Algumas tarefas são processadas em módulos separados e não se interferem. Outras tarefas requerem processos simultâneos em únicos módulos e são desta forma difíceis de serem executadas em paralelo. Esta organização modular será um ponto discutido neste trabalho, mais especificamente no capítulo referente a Redes Neurais Hierárquicas, pois, serve de base para a arquitetura proposta. Também um ponto de indagação é o aparecimento desta organização. Quando, no desenvolvimento do cérebro, esta organização modular surgiu?

Um reforço para a visão de que parte desta organização já está pré-ligada, são as diferentes localizações de algumas regiões funcionais no cérebro e que a localização da maioria das funções estão localizadas no mesmo lugar em quase todos os indivíduos. Um exemplo que também apóia esta visão: a maioria das fases no desenvolvimento da linguagem, por parte das crianças, é a mesma para qualquer idioma.(NIYOGI, 1995)

2.4 Noções de Etologia

Os cientistas Karl Von Frisch e Konrad Lorenz, austríacos, e Nikolas Tinbergen, holandês, ganharam em 1973 o prêmio Nobel da Medicina e Fisiologia por terem criado uma nova ciência, a Etologia, que faz o estudo comparado do comportamento dos animais. Mas, o que vem a ser o comportamento animal? As definições são difíceis, pois o comportamento é uma das atividades pelas quais um animal se mantém. (MESSENGER, 1980). Engloba todas as atividades como procurar comida, evitar perigo, acasalar-se e até criar sua prole. Isto requer a ativação dos órgãos de sentido que coletam informações, principalmente do mundo externo. Devemos ater-nos ao sistema muscular, pois é através da sua atividade ou inatividade, diante de uma situação, que determinamos ou caracterizamos o comportamento.

O comportamento de um animal também depende da situação em que ele se

encontra. Podemos citar muitos exemplos de comportamento usando nossa própria experiência de vida. Então, por exemplo, se estamos caminhando e em nossa direção vem uma outra pessoa, nossa reação (comportamento) é diferente dependendo de quem está vindo, pois, se for uma pessoa desconhecida iremos desviar para não nos chocarmos. Porém, se for uma pessoa conhecida, poderemos ir ao seu encontro para cumprimentá-la. Se estivermos diante a um fogão, onde estamos esquentando algo em uma jarra de metal, sabemos que após algum tempo não podemos pegar diretamente na parte de metal, pois iremos nos queimar.

Se estivermos diante de um fogo e algo importante para nós cair nele, podendo estragar, nós sabemos que se tentarmos pegar rapidamente não nos queimaremos. Este tipo de comportamento já estava previamente guardado em nossa memória, pois de alguma forma nós aprendemos, por vivência própria ou porque alguém nos relatou algo parecido.

Este exemplo nos mostra bem que deve existir uma hierarquia de controladores dentro do sistema nervoso, pois o primeiro comportamento, ao segurarmos um objeto muito quente, é o de largá-lo para não nos queimarmos. Entretanto, se este objeto for muito valioso ou raro, um controle de nível superior é acionado e inibe o comportamento de largar imediatamente o objeto, acionando então um comportamento que nos faça procurar um local para largar o objeto em segurança.

Um fator de complicação vem do fato dos comportamentos não serem somente aprendidos por experiência ou ensinados. Grande parte dos comportamentos já está “pré-fiados” no sistema nervoso central, e foram determinados por processos evolutivos, isto é, determinados por genes.

Se pegarmos como exemplo a espécie de gaivota que para proteger sua prole de predadores, retiravam as cascas dos ovos já chocados do ninho e levavam para longe, poderíamos dizer que no passado pode ter existido, dentro desta mesma espécie, algumas que não retiravam as cascas e, por isto, suas proles não chegavam a idade de reprodução, resultando no seu desaparecimento, como mostrado em DAWKINS (1989).

Então, aquelas que sumiram eram diferentes geneticamente dos tipos bem

sucedidos, portanto, a crença de que o comportamento evoluiu leva-nos a supor que houve uma variação genética. Somente neste sentido podemos dizer que o “comportamento foi determinado por genes”.

Mesmo podendo dizer que o comportamento foi determinado geneticamente, há a complicação adicional na qual mostra-se que o ambiente contribui para a variação. Tem-se que avaliar a maneira como o indivíduo foi tratado desde a sua infância até a fase de reprodução (adulto). Isto afeta o comportamento tanto quanto a linhagem genética a que pertence. Podemos perceber que o comportamento pode ser determinado por genes e também pelo ambiente em que vivemos.

2.4.1 Classes de Comportamentos

2.4.1.1 *Comportamentos Reflexivos*

Uma característica parece ser marcante tanto aos comportamentos reflexivos quanto às taxias refere-se ao fato de que a intensidade e a duração das respostas dos organismos são uma função direta da intensidade e duração do estímulo que disparou o comportamento. Também é importante notar que o tempo decorrido entre a ocorrência do estímulo e o aparecimento da resposta é mínimo. O arco reflexo, em uma forma simples como em um reflexo proprioceptivo, pode envolver apenas dois neurônios, o aferente (sensorial) e o eferente (motor). Estes reflexos são chamados de monossinápticos pelo fato da mensagem ter que transpor apenas uma sinapse.

É óbvio que este tipo de resposta é adaptativa e anula uma entrada nociva com um retardo mínimo. Reflexos do tipo flexor e extensor cruzado necessita de interneurônios e mais de uma sinapse, mas eles também são atos simples e breves.

Em animais como os vertebrados ou insetos, o arco reflexo não passa de uma abstração de alguma utilidade, mas realmente não é tão significativo para uma descrição detalhada. Com certeza explica o comportamento de um animal pisando em um espinho, mas não descreve, por exemplo, a inter-relação entre chipanzés ou o processo de aprendizagem da linguagem em humanos. Entretanto, em animais com sistema nervoso

contendo menos neurônios, este reflexo pode abranger todo o comportamento, isto é, o reflexo é o próprio comportamento.

2.4.1.2 *Comportamentos Reativos*

É uma classe intermediária entre os comportamentos puramente reflexivos e os comportamentos instintivos. É formada por uma série de comportamentos estereotipados como resposta a um dado estímulo. O estímulo que dispara o comportamento é geralmente mais complexo e específico que o necessário para disparar um comportamento reflexivo. A resposta envolve uma sequência temporal de ações que se desenrolam até o seu final, mesmo o estímulo disparador não estando mais presente.(ROISENBERG, 1996)

Um exemplo deste comportamento é uma espécie de ganso que ao ver que seu ovo não está no ninho, vai até o mesmo, encaixa-o no meio de suas pernas para rolá-lo até o ninho. Durante o trajeto, se porventura, o ovo cair e este fato não for percebido pelo ganso, o mesmo continua até chegar ao ninho e só neste momento, ele irá perceber que o ovo não está ali. Então, terá que voltar, pegar novamente o ovo e repetir o processo.

Outro exemplo é o caso de uma certa espécie de vespa que, após emergir de sua pupa, terá poucas semanas para realizar sua tarefa: acasalar, construir seu vespeiro, caçar suas presas para armazenar no vespeiro, botar os ovos, fechar e esperar sua morte. Estes processos são desencadeados pelo encontro da vespa fêmea com o zangão (vespa macho). Se, por ventura, algum estímulo não for disparado a sequência estará comprometida.

3 ALTERNATIVAS DE IMPLEMENTAÇÃO PARA AGENTES AUTÔNOMOS

As implementações que aqui serão citadas nos serviram de base para o desenvolvimento da rede hierárquica, pois elas trabalham com comportamentos reflexivos, no caso dos veículos de Braitenberg, e com “memória”, no caso da arquitetura proposta por BROOKS (1990b) e por ALVES (1993).

Além das implementações que serão citadas, existem outras formas de implementações de agentes autônomos, assim como, há diferentes tarefas que eles executam. Um exemplo disto pode ser visto em HANCOCK (1999). Os AA's têm suas próprias características, inspiradas em alguma outra estrutura computacional ou até mesmo em seres vivos.

Uma das características que podemos encontrar em muitas pesquisas nesta área é a tentativa de criar agentes autônomos que tenham a nossa aparência, tanto física quanto na maneira de pensar e agir. Enquanto não achamos a fórmula certa para isto, vamos criando AA's que tenham outras aparências, mas com alguns comportamentos parecidos com o nosso.

Estes AA's podem ter a forma de um tambor e serem implementados através de redes neurais artificiais que são baseadas na rede neural dos animais. Como também, podem ser implementados através de algoritmos genéticos, que não deixam de ser inspirados na Natureza e seu processo de Evolução.

Como podemos ver, ainda estamos intimamente ligados à Natureza, uma vez que nos baseamos em seus indivíduos ou na sua maneira de produzi-los para podermos criar seres mecânicos dotados de inteligência.

Até os meados da década de 80 as alternativas de implementações não eram capazes de operar em tempo real no mundo real, já que estas implementações eram baseadas em programação direta e IA simbólica. A ineficácia destas implementações veio a ser superada com o aparecimento de novas propostas; um dos pioneiros foi

BROOKS (1985).

Em seu artigo, “Elephants Don’t Play Chess”, BROOKS (1990b) descreve uma forma de implementação chamada de “Subsumption Architecture”. A partir desta proposta viu-se a possibilidade de uma virada no desenvolvimento da Inteligência Artificial, pois ela não se baseia unicamente na representação simbólica do conhecimento para a construção de AA’s capazes de operar no mundo real. Muitos pesquisadores, como FERRELL (1993), seguem esta forma de implementação.

Estas novas propostas buscavam diretamente no mundo físico a representação para o desenvolvimento dos chamados sistemas reativos ou comportamentais (“behavior-based”). Não existe explicitamente dentro destes sistemas um objetivo, plano ou modelo do mundo, ocorrendo apenas uma reação à situação que eles têm em mãos.

A capacidade de operar em ambientes desestruturados, que podem ser modificados dinamicamente, foi a nova conquista efetuada pelos robôs autônomos, através desta abordagem de BROOKS (1990b), e que foi chamada por ALVES (1993) de “Arquitetura de Assunção”. Esta arquitetura pode ser encontrada detalhada em ALVES (1993). Um trecho será reproduzido a seguir:

“A abordagem baseada em comportamento conseguiu, de fato, fazer avançar a pesquisa envolvendo o paradigma de robôs de terceira geração, ou robôs inteligentes. Um controlador baseado em comportamento é dividido em camadas. Cada camada é responsável por um tipo de comportamento ou tarefa a ser executada pelo sistema global. Nesta estrutura paralela, cada camada possui um caminho completo dos sensores aos atuadores; a arquitetura de controle permite a produção de comportamento útil, antes de todo o conjunto de camadas, inicialmente projetado, estar completo; a estrutura em paralelo reduz a tendência ao congestionamento das informações; a falha de qualquer das camadas não significa, necessariamente, o colapso total do sistema. Cada camada usa apenas as informações do sensor que lhe é acoplado e a percepção de

mundo que necessita para sua tarefa específica, isto é, cada camada é completa (no sentido de implementar um comportamento ou tarefa). Cada módulo, então, precisa resolver somente o problema que lhe é pertinente. Brooks construiu um robô móvel autônomo utilizando a arquitetura de “assunção” cuja primeira camada, a camada 0, evitava obstáculos. Em seguida, adicionou a camada 1 que introduz a atividade de fazer com que o robô se dirija a um determinado lugar. Independentemente, a primeira camada livra o robô dos obstáculos que se lhe interponham. A segunda monitora o progresso do robô e envia comandos atualizados aos atuadores na perseguição de seu objetivo, sem estar consciente, explicitamente, dos obstáculos que foram evitados pela camada inferior.

Cada camada é composta por uma rede de topologia fixa de máquinas de estado finito simples que são combinadas através de mecanismos chamados inibidor e supressor. Cada máquina de estado finito possui uma quantidade de estados, um ou dois registradores internos, um ou dois relógios internos e acesso às máquinas simples que podem fazer cálculos tais como soma de vetores. São ativadas através de mensagens que recebem, e uma mudança de estado ocorre quando chega uma determinada mensagem ou quando o tempo estipulado para este estado expira. Não há memória global compartilhada. As entradas de cada máquina de estado finito podem ser suprimidas e as saídas podem ser inibidas por outras máquinas.” (ALVES,1993).

Para mostrar graficamente esta proposta, foram extraídas as figuras a seguir, do artigo de ALVES (1993):

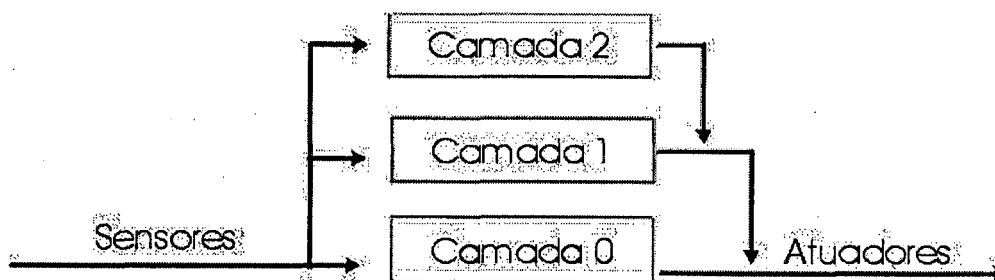


Figura 3.1: Arquitetura em camadas de comportamentos da “Subsumption Architecture”.

A figura a seguir mostra o Esquema Inibidor/Supressor na camada de comportamento:

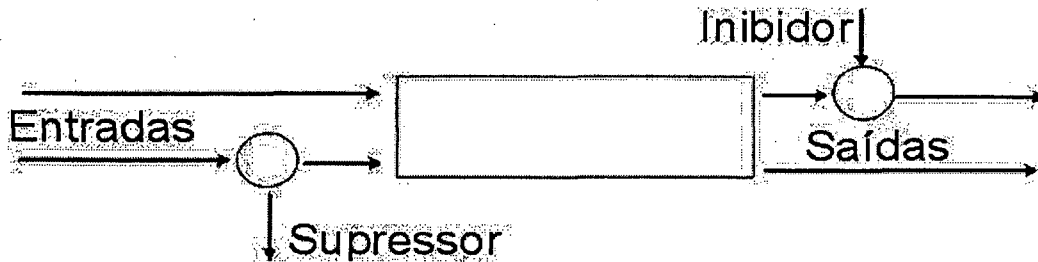


Figura 3.2: Esquema Inibidor/Supressor para camada de comportamento.

Evitar uma modelagem simbólica do mundo real é uma meta que esta arquitetura procura evitar por se tratar de uma tarefa difícil. Ela, também, não nos deixa com total segurança em alcançar problemas que requeiram planejamento complexo, além de ter uma série de limitações e dificuldades, como as listadas a seguir:

- Um dos problemas é ter que codificar manualmente os autômatas de estado finitos, expandidos com elemento de temporização, antes de implementá-los. Eles foram usados para implementar cada módulo de comportamento.
- A adaptação a outras tarefas fica muito difícil, uma vez que, será difícil a reconfiguração do sistema, pois, existe uma interdependência entre os comportamentos implementados.

Alguns AA's implementados com a "Subsumption Architecture" estão descritos a seguir:

Agente Subsumption: Allen

Foi o primeiro robô desenvolvido por Brooks. Ele tem um sensor de distância (sonar) e um hodômetro "onboard". E "offboard" uma máquina Lisp para simular três níveis da arquitetura. A primeira camada permite que o robô não fique imóvel e também evite obstáculos. A segunda camada é responsável por fazer o robô vagar aleatoriamente. A terceira camada faz o robô "olhar" (com seu sonar) para locais distantes e tenta ir em direção a eles. O progresso é monitorado pelo hodômetro.

Agente Subsumption: Attila

Attila é um robô com seis pernas e tem 150 sensores, 23 atuadores e 11 processadores “onboard”. Ele é capaz de ter comportamentos complexos e impressionantes. Sua particular função é de exploração planetária autônoma.

Agente Subsumption: Herbert

O robô Herbert é fisicamente mais ambicioso do que os dois citados acima. Ele tem um computador com 24 processadores “onboard”, livremente distribuídos para executar a Subsumption Architecture. Tem 30 sensores de proximidade (infravermelho) e consegue vagar pelos corredores de escritórios, entre pessoas e objetos, a procura de latas de refrigerantes para recolher (Não faz menção sobre identificar entre latas cheias e vazias). Seus comportamentos são: desviar de obstáculos, seguir paredes, reconhecer em tempo real latas e objetos como escrivatinhas e tem um grupo de 15 comportamentos para direcionar o braço para pegar uma lata.

Agente Subsumption: Tom and Jerry

Tom e Jerry são dois robôs idênticos com uma “Subsumption architecture” de três camadas construída para demonstrar como uma implementação pequena e grosseira é necessária. O programa todo foi implementado em um 256 gate PAL. Eles são dois carros com um sensor infravermelho, porém, um tem o sensor na parte frontal e o outro na parte traseira. Os comportamentos são simples, pois o robô com o sensor frontal é atraído por objetos móveis e o outro “foge” de “quem” se aproxima dele. Colocando os dois, o Tom (sensor frontal) de frente para a traseira do Jerry (sensor traseiro), o comportamento emerge quando o Jerry detecta o Tom e se distancia, causando a detecção por parte do Tom, que ao detectar vai atrás.

Os robôs descritos acima são alguns dos exemplos que trazem a implementação da arquitetura proposta por Brooks, porém não foram as únicas implementações a nos inspirar para a criação da proposta que será apresentada neste trabalho. Uma outra alternativa foi proposta por HOGG et al (1991), baseada nas criaturas de Braitenberg,

onde, ele descreve a implementação de vários AA's, cada um com seu objetivo, mas nenhum destes AA's com grande capacidade "intelectual".

Neste artigo os AA's são "robôs" montados através dos "tijolos" da LEGO, com os quais se pode configurar várias formas, inclusive animais. Os comportamentos obtidos com estes AA's são do tipo reflexivo, como seguir luz ou desviar de obstáculos.

As primeiras seis "criaturas", como são chamadas por Braitenberg, são bastante simples e seus nomes são: "Timid", "Indecisive", "Paranoid", "Dogged", "Insecure", e "Driven".

O "Timid" (Tímido) tem na sua configuração apenas um tijolo motor e um sensor de luz que aponta para cima. Ele tem como objetivo andar quando o sensor de luz detectar um excesso de luminosidade, isto é, quando a luz está em excesso, o motor é acionado e o "carro" anda para frente. Seu comportamento se resume em ir andar até que encontre a uma sombra e quando entra na sombra ele fica parado. Então, se detecta novamente uma luz forte, ele andará, mas se no meio do caminho esta luz for apagada ou se algo tapar o foco, ele irá parar.

A outra criatura é o "Indecisive" (Indeciso). Sua configuração é idêntica ao do Tímido, diferenciando apenas nos motores, que assumem duas situações distintas. Estas situações determinam para onde o veículo deve se mover, isto é, quando a entrada é ON (ligada) o veículo vai para frente, isto também significa que ele encontrou luz, no entanto se a entrada receber OFF (desligada) o veículo inverte os motores e dirige-se para trás. Esta situação indica que a luz foi apagada ou ele entrou em uma sombra.

Imagine a seguinte situação:

- O Robô detecta um foco luminoso e vai em sua direção, porém no meio do caminho uma sombra faz com que seus sensores deixem de detectar o foco, com isto, os motores são invertidos e ele anda para trás. Andando para trás, fará com que ele perceba novamente o foco, então, reverte novamente o motor e vai em frente até encontrar a sombra. Estes movimentos ficariam se repetindo. Digamos que é um comportamento não

muito inteligente.

A terceira criatura é o “Paranoid” (Paranóico). Este já assume uma forma de animal, no caso, uma tartaruga. Ele tem dois motores, um do lado esquerdo e outro do lado direito, acionando cada um a sua roda.

Assim como as criaturas “Tímido” e “Indeciso”, o “Paranóico” tem seu sensor de luminosidade. O sensor é “ON” (ligado) quando detecta uma luz e “OFF” quando está na sombra. Estas situações levam o veículo a ter comportamentos interessantes.

Por exemplo, se ele estiver detectando um foco de luz, conseqüentemente, irá em sua direção, entretanto se por algum fator ele acaba encontrando uma sombra, o motor esquerdo reverte. Isto causará uma virada para a esquerda, pois, somente o motor esquerdo reverte, o direito continuará indo em frente. A figura, extraída do artigo de HOGG et al (1991), está a seguir para demonstrar este comportamento:

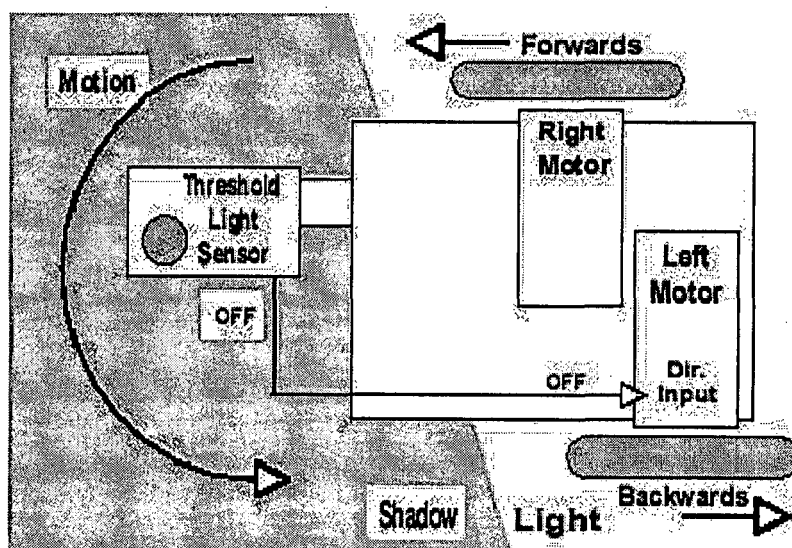


Figura 3.3: Criatura “Paranóico”

O “Dogged” (Teimoso ou Obstinado) tem como configuração apenas um tijolo com motor e roda, e um sensor de toque que está ligado a dois “botões”, um na parte frontal e outro na traseira. Estes botões estão colocados em um tipo de pára-choques.

Seu comportamento depende do toque em um dos seus pára-choques, pois se o toque for no pára-choque dianteiro ele reverte o motor indo para trás e,

conseqüentemente, se o toque for no pára-choque traseiro o motor reverte, impulsionando o veículo para frente.

O quinto veículo é o “Insecure” (Inseguro). Ele usa um tipo de pêlo, antenas, se é que podemos chamar assim, para detectar o que está ocorrendo em sua vizinhança. Ele tem a aparência de uma tartaruga, assim como o “Paranóico”, mas sua tarefa é identificar, por exemplo, uma parede e ir tateando-a.

Se este veículo estiver em um local aberto o suficiente para não detectar esta parede, ele ficará girando em círculos, caso contrário, irá direto para a parede. Também não é um comportamento que consideremos de grande funcionalidade.

A última criatura simples é o “Driven” (Dirigido ou Guiado). Sua principal tarefa é a de seguir a luz. Seus sensores de luminosidade estão invertidos, isto é, o sensor posicionado à esquerda está ligado ao motor do lado direito e, conseqüentemente, o sensor do lado direito está ligado ao motor da esquerda. Então, quando o sensor da direita for acionado, o motor da esquerda irá avançar e vice-versa.

HOGG et al (1991) neste artigo também citou quatro outras criaturas, um pouco mais complexas que as anteriores. Mas seus comportamentos não deixaram de ser reflexivos, no entanto, em alguns casos dois veículos passaram a executar a tarefa.

O primeiro veículo é chamado de “Persistente”, sua configuração é parecida com a do “Driven”, sendo mudada apenas, pela presença de um sensor de toque. Ele também irá de encontro à luz, mas este sensor de toque terá a incumbência de desviar de obstáculos.

Uma diferença também aparente neste veículo é, se ele, durante sua procura por luz, tocar em qualquer coisa considerada como obstáculo, ele reverterá o motor, contudo isto será controlado por um temporizador. Então, se ocorre dele se chocar com um obstáculo enquanto estiver indo em direção ao seu objetivo, a luz, ele anda de ré por um tempo, porém retoma o mesmo caminho em direção à luz, significando que irá se chocar com o mesmo obstáculo.

É um comportamento que não podemos considerar muito inteligente. A diferença em relação às outras criaturas é o temporizador, pois este comandará a volta ao caminho e não é necessário um choque traseiro para ele andar para frente novamente.

A segunda criatura considerada complexa, na realidade, é formada por uma dupla de criaturas. Elas são as criaturas chamadas de “Atrativa” e “Repulsiva”.

As duas trabalharão em conjunto para a execução da tarefa que é procurar luz e fugir da colisão. Isto ocorrerá devido a uma procurar a luz e a outra fugir da colisão, porque um sensor ao foco de luz será acoplado à atrativa, e um foco de luz à Repulsiva.

Então, imagine a seguinte situação: os dois veículos estão em uma linha reta e a criatura Repulsiva está de frente para a Atrativa. A Repulsiva vai aproximando-se lentamente até o sensor de luminosidade da Atrativa detectar a luz. Neste momento a Atrativa sai rapidamente, afastando-se da Repulsiva e ficam nesta “brincadeira”, pois mesmo afastando-se, a Repulsiva detecta a luz acoplada à Atrativa e novamente encaminha-se para ela.

A “Consistente”, como é chamada, é uma outra criatura um pouco mais complexa na sua configuração. Sua aparência também é de uma tartaruga. Seu sensor é diferente dos demais, já que agora se trata de um sensor de sons.

Através de um microfone acoplado, este veículo detecta sons e vira-se para o lado onde foi produzido este som. Na realidade, ele tem 4 estados: “ON-ON”, “OFF-ON”, “ON-OFF”, “OFF-OFF”. Estes estados são referentes a dois “tijolos” de FLIP-FLOP, responsáveis por ativar ou desativar o motor.

O primeiro estado “ON-ON” faz com que o veículo se movimente para frente. O segundo “OFF-ON” é responsável por virar o veículo para a esquerda. O terceiro estado “ON-OFF” faz com que o veículo vire para a direita e o último estado “OFF-OFF” tem a responsabilidade de movimentar o veículo para trás. Então, se alguém bater palmas repetidamente, ele tentará ir de encontro a este barulho.

A quarta e última criatura não é um “animal” ou “veículo”, trata-se de uma

ratoeira “inteligente”, chamada de “Inhumane” (Inumano ou desumano). Através de sensores e atuadores, ela irá prender um “rato”. Sua configuração seria, mais ou menos, uma caixa em formato de um corredor comprido e chato.

Na extremidade fechada é colocada uma isca e um pouco antes, dois furos, um de cada lado da caixa, onde é colocado, de um lado, um foco de luz e, do outro, um sensor de luminosidade. No início da caixa é colocado um tipo de porta motorizada, a qual está ligada ao sensor de luminosidade.

Teoricamente o que ocorrerá é a entrada de um rato, para pegar a comida (isca), e assim que ele passar pelo foco luminoso, irá interromper a ligação entre o sensor e o foco, com isto, será acionada a porta, fechando a extremidade aberta e trancando o rato dentro.

A figura extraída do artigo está a seguir, para mostrar graficamente como funciona esta ratoeira:

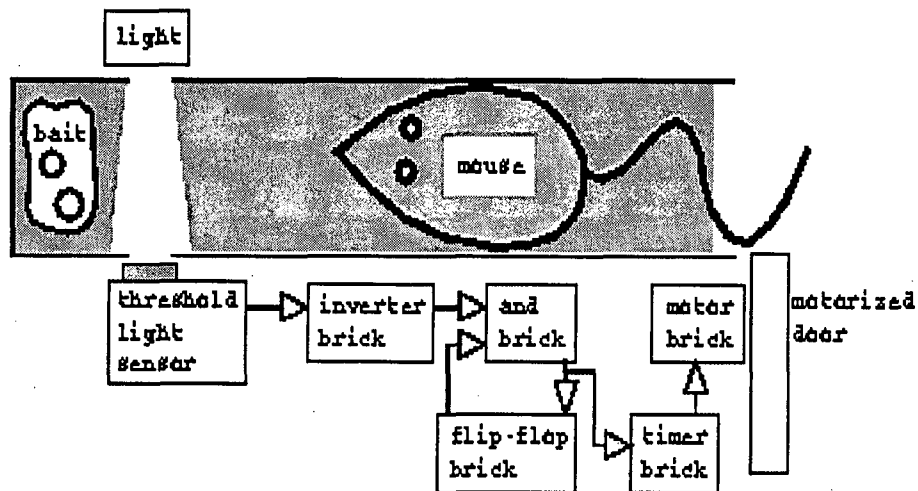


Figura 3.4: “Inhumane” criatura Inumano.

Com isto foram apresentadas algumas implementações alternativas de AA's móveis. Podemos perceber destas implementações que os comportamentos implementados são sempre reflexivos, isto é, básicos e sem muita complexidade.

Apesar disto, foram de grande importância para o desenvolvimento da proposta feita neste trabalho, principalmente a arquitetura de Brooks, com relação às camadas de

comportamentos que nos dá a idéia de uma hierarquia, onde os comportamentos mais simples são implementados nas camadas inferiores e comportamentos mais complexos são implementados em camadas superiores.

Não menos importante foi o trabalho de Hogg, Martin e Resnik que implementaram os comportamentos reflexivos baseados nas criaturas de Braitenberg. Também nos deram inspiração para a implementação dos comportamentos reflexivos da arquitetura hierárquica desenvolvida neste trabalho. No capítulo referente à nossa proposta, capítulo 5, poderá ser visto a relação entre a proposta de Brooks e a de Hogg, Martin e Resnik.

4 CONTROLE HIERÁRQUICO DOS COMPORTAMENTOS DE UM AGENTE AUTÔNOMO

Para que pudéssemos verificar a validade das idéias e objetivos que pretendíamos atingir, realizamos uma implementação prática que será mostrada a partir desta seção. A implementação foi inserida em um sistema de domínio público que é o "Khepera Simulator", Anexo A, desenvolvido por Olivier Michel.

Este trabalho foi realizado seguindo algumas etapas, onde a primeira consistiu em montar um ambiente com alguns obstáculos e um foco luminoso para avaliar o desempenho do robô simulado. Este ambiente teve uma composição relativamente fácil (somente paredes) e aumentou-se a sua complexidade conforme os resultados foram sendo obtidos.

Para a implementação dos comportamentos no robô simulado, fez-se necessário a identificação dos valores dos seus sensores, pois através deles determinamos o seu "comportamento". O posicionamento do robô no "mundo", se ele está perto ou distante de um obstáculo ou de um foco luminoso, é dado pelos sensores.

A etapa seguinte foi a implementação de uma hierarquia de rede neural capaz de selecionar e coordenar o comportamento do robô fazendo-o "desviar de obstáculos" e "ir de encontro a um foco luminoso". Podemos compará-lo ao comportamento de uma criança aprendendo a andar, uma vez que, terá que se manter em pé e, ao mesmo tempo, desviar de obstáculos para não se machucar.

Após a verificação de que o robô conseguiu desviar de obstáculos e também foi ao encontro de um foco luminoso, assim que o detectou, modificamos a hierarquia criando mais comportamentos reflexivos, reativos e um controle de energia. O intuito foi conseguir fazê-lo seguir paredes enquanto sua energia estivesse normal.

E fazê-lo "perceber" quando sua energia chegou a um nível baixo a ponto dele ter que percorrer o mundo em busca de um ponto de energia para reabastecer suas "baterias".

Isto significa que os comportamentos reflexivos são controlados pelos comportamentos reativos, representados por redes recorrentes implementadas em um nível acima, dos comportamentos reflexivos, na hierarquia. E em um nível mais acima foi implementado o controle do nível de energia do robô.

4.1 Comportamentos Reativos e Reflexivos

4.1.1 Rede Neural Artificial Hierárquica

A implementação de AA's pode ser feita com vários tipos de sistemas, como sistemas especialistas, redes neurais, etc. Mas até que ponto? Ou qual é o grau de inteligência alcançado por estes sistemas?

O problema de medir a inteligência de máquinas foi proposto por L. A. Zadeh na década de 90. Assim como a medida de inteligência humana é assunto de fecunda pesquisa, bem mais antiga, mas ainda atual, envolvendo QI (Quociente de Inteligência) e IE (Inteligência Emocional), etc. A inteligência de máquina é assunto em aberto. Recentemente FALQUETO et al. (2001) abordaram o problema usando inspiração biológica.

Como podemos encontrar uma arquitetura que possibilite um grande repertório de comportamentos, e ainda, possa fazer a seleção e coordenação do comportamento a ser ativado a cada instante? Em MEDEIROS (1998) encontramos as seguintes arquiteturas:

Arquitetura NASREM:

Na arquitetura NAS-REM "NASA/NBS standard reference model" as informações percebidas passam por várias fases do processo até uma visão coerente da situação atual ser obtida.

Depois disso, um plano é adotado e sucessivamente decomposto por outros módulos até que a ação desejada possa ser diretamente executada pelos atuadores.

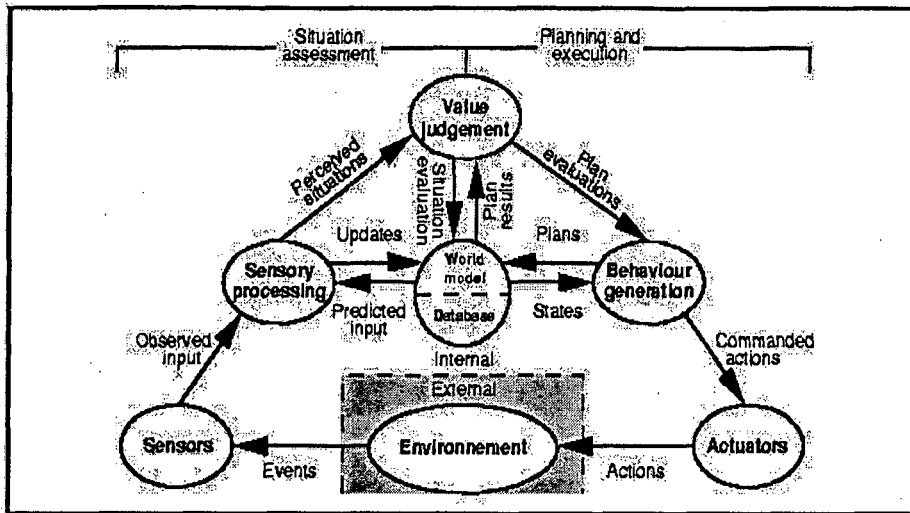


Figura 4.1: Arquitetura NASREM - NASA/NBS standard reference model-
Journal of the Brazilian Computer Society

Arquitetura TCA:

A arquitetura TCA (Arquitetura de Controle de Tarefas) é um sistema operacional de robô de alto nível com um grupo integrado de mecanismos comumente precisos para apoiar comunicações distribuídas, decomposição de tarefa, administração de recurso, execução supervisionada e recuperação de erro.

Um sistema usando TCA consiste em um número de módulos de tarefas específicas e um propósito geral de reutilizar o módulo de controle central. Os módulos comunicam com os outros (e com o controle central) passando mensagens.

Os módulos registram as mensagens negociando processos com o TCA e o controle central tem responsabilidade por fazer o roteamento de mensagens para os módulos apropriados.

As arquiteturas de TCA retêm muitos conceitos das arquiteturas de quadro-negro, mas difere deles porque enquanto TCA mantém as informações centralizadas no controle, os dados atuais necessários para resolver problemas são distribuídos entre os processos do sistema.

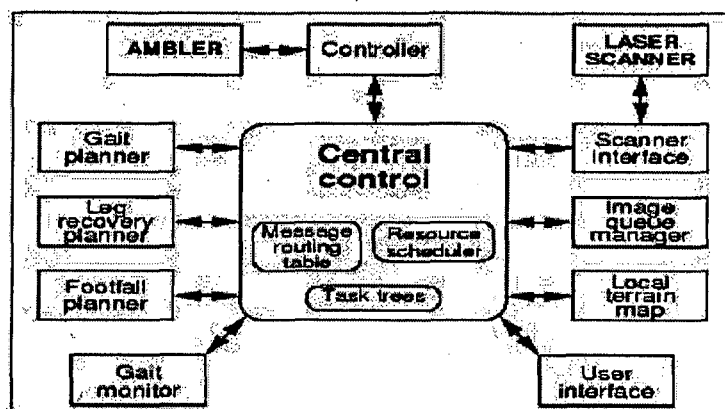


Figura 4.2: Arquitetura TCA (Task Control Architecture)-
Journal of the Brazilian Computer Society

As duas arquiteturas apresentadas não mostram que existe uma hierarquia, mas apenas informações sendo trabalhadas por módulos que transmitem essas informações para um controle central e esse transmite para os atuadores.

Na busca pela arquitetura ideal para conseguirmos um alto grau de inteligência e autonomia para os AA's, nos deparamos com a solução encontrada pela Natureza, assim como aparece nos animais, que foi a de criar uma arquitetura neural hierárquica.

Então, buscando inspiração na Natureza para implementações de controles eficazes, foi possível perceber que existe uma arquitetura no sistema nervoso dos animais, e essa arquitetura é hierárquica. Verificamos sua existência através dos estudos da medula, controlador de alguns comportamentos, e nos estudos do cérebro, controlador de outros comportamentos mais complexos.

O nível de inteligência é diferente para os seres vivos, todavia todos são dotados de um grau de inteligência e cada um voltado à sua função em específico. Alguns exemplos foram citados no capítulo 2.

Quando falamos em implementação de AA's com alto grau de inteligência, a pretensão é a de chegarmos a um AA com comportamentos semelhantes aos dos humanos. Para tentarmos isto temos que levar em conta o equipamento (hardware) necessário para atingir este grau. Então, vamos analisar da seguinte forma - um ser humano tem aproximadamente 10^{11} neurônios, e cada um faz cerca de 10^2 a 10^4

sinapses e pode receber informação de aproximadamente 10^3 outros neurônios (BARRETO, 2000), (KOVÁCS, 1997).

Antes de tentarmos atingir o grau de inteligência de um Humano, poderíamos, primeiro, tentar atingir o grau de inteligência de uma barata ou de um rato, visto que eles têm comportamentos inteligentes. Por exemplo, uma barata sabe quando “um predador” está perto ou quando está em perigo, pois ela sente através de seus cercis (um tipo de biruta na sua parte traseira) se um deslocamento de ar é diferente do normal, indicando um suposto perigo. A barata tem neurônios especializados em detectar esta variação no ar ao seu redor.

Para atingir um primeiro estágio nesta busca, devemos implementar um comportamento básico, como evitar colisões ou até mesmo, como no caso das baratas, dar a eles a capacidade de sentir a presença de algo ao seu redor.

Nestes casos acima, bastaria a implementação de uma rede neural direta capaz de controlar estes comportamentos básicos reflexivos. Após este objetivo ser alcançado, poderia ser usado um dos princípios da Natureza, para aumentar o grau de complexidade dos comportamentos, a Evolução. A evolução do AA se daria através da implementação de outros comportamentos básicos.

Mas para coordenar esses comportamentos reflexivos de modo que cheguemos em comportamentos mais complexos, por exemplo, comportamentos reativos, torna-se necessário a implementação de uma entidade superior interconectada às redes diretas de maneira modular e hierárquica. Para essa categoria de comportamento devemos atribuir outra topologia de rede neural, a rede recorrente. Com esta evolução chegamos a AA com comportamentos mais complexos, com uma estrutura de controle, destes comportamentos, mais complexa.

Nossa primeira proposta diz respeito ao controle de comportamentos do tipo reflexivo e reativo, na qual as redes diretas serão responsáveis pela implementação de comportamentos reflexivos básicos e, as redes recorrentes implementarão as tarefas de coordenação e seleção destes comportamentos.

A intenção é obter, como resultado final, o controle refinado do comportamento dos AA's através da hierarquia de redes neurais artificiais. Como citado em MESSENGER (1980), iremos controlar o controlador. Também podemos encontrar uma arquitetura baseada em hierarquia em CARPINTEIRO (1997). Um outro autor que apresenta um trabalho visando mostrar uma arquitetura hierárquica é SMIEJA(1991): "Multiple Network Systems (Minos) Module: Task Division and Module Discrimination".

Entendemos que os comportamentos de um AA em relação aos seus atuadores são, então, reflexivos e de certa forma ineficazes em alguns momentos, pois este tipo de comportamento não requer memória. Supondo que um AA tivesse como missão apenas andar por um determinado "mundo", e neste mundo existissem objetos para atrapalhá-lo, é muito provável que este AA encontraria situações em que ficaria circulando por rotas já percorridas, ou melhor, poderia encontrar "becos" onde ficaria andando em círculos. Esta situação indicaria a falta de reconhecimento de trajetória.

Assim como o SNC humano é modular, existem sistemas computacionais que também são e criar estes sistemas modulares traz algumas vantagens, como favorecer um comportamento robusto e flexível (BUSSLER & URBAN, 1997).

Existe nesta arquitetura uma modularização, já que cada rede do subsistema sensório trabalha isoladamente a sua informação e passa, após o processamento, para a rede de controle, que também trabalha isoladamente a informação. A rede de controle transmite a informação para as redes motoras que irão operar diretamente sobre os atuadores. Olhando cada rede separadamente vemos que são módulos trabalhando uma informação e transmitindo após obter um resultado interno, mas, olhando o sistema como um todo, vemos que os módulos trabalham em conjunto para obter um resultado final que chamamos de comportamento. Na configuração "horizontal" do primeiro nível, nível mais baixo, tem-se as redes diretas do subsistema sensório e do subsistema motor. A rede de controle se encontra em um nível acima das redes do subsistema sensório e motor. Olhando a configuração "vertical" temos a rede recorrente (rede de controle), responsável pelos comportamentos reativos, logo acima das redes diretas,

como mostra a fig. 4.3.

Esta configuração contém quatro redes no nível mais baixo e uma rede no nível acima, no entanto, podem ser várias redes no nível de comportamentos reflexivos e também no nível de comportamentos reativos, dependendo da complexidade que queremos dar ao AA.

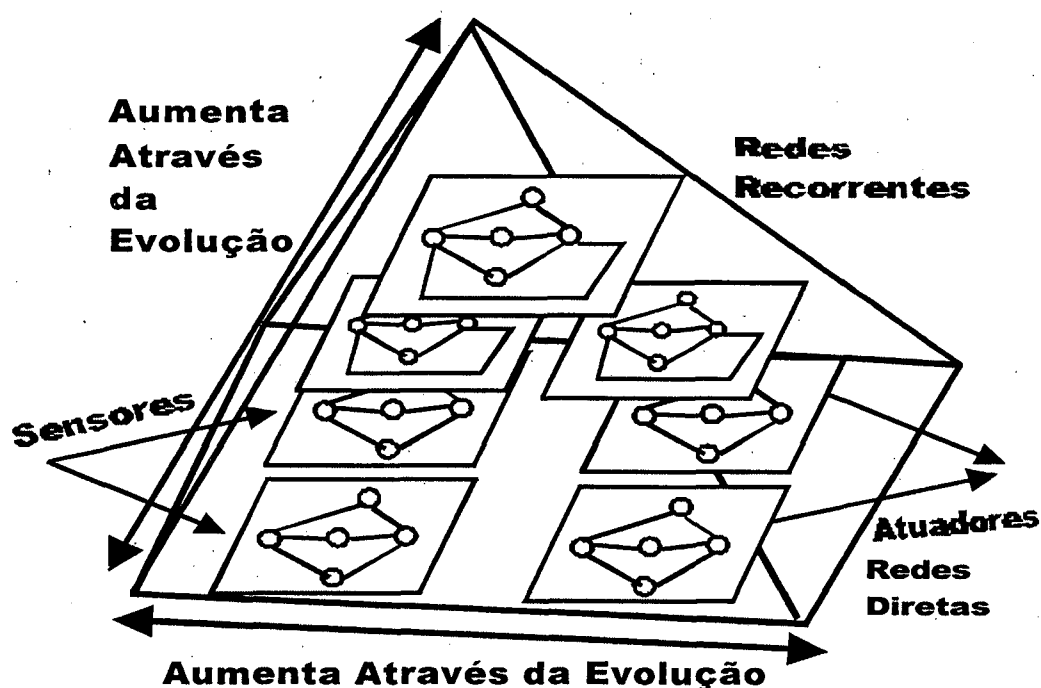


Figura 4.3: Rede Neural Hierárquica proposta para o controle dos comportamentos.

O esquema da fig. 4.3 visa mostrar que as redes de controle, níveis superiores da hierarquia, fazem o controle das demais. Pode-se ter mais redes controlando e tornando cada vez mais complexa a rede de controle do comportamento em AA's, como mencionado acima.

4.1.2 Prioridade de Comportamento

Visto que os seres vivos têm algumas categorias de comportamentos, como por exemplo, reflexivos e reativos, existe a necessidade de coordená-los. Esta coordenação não será apenas no sentido de separação dos tipos de comportamentos, mas terá que determinar qual comportamento deverá emergir num dado instante.

Isto significa que um comportamento terá prioridade sobre outros, pois, naquele instante só poderá ser tomada uma decisão e não poderá haver conflitos de comportamentos, o que poderia resultar em perigo para o indivíduo.

No AA (Robô simulado), onde implementamos esta arquitetura, também existirá esta prioridade de comportamentos, uma vez que ele não poderá ficar parado por não saber qual “atitude” tomar. Esta prioridade é bastante necessária para nós e também para os outros animais, por exemplo, uma pessoa que possui uma alergia a um alimento, que não seja muito grave, e sem saber disto ingere este alimento, com certeza, depois de algum tempo esta alergia vem a tona.

Numa próxima vez, estando diante deste alimento, esta pessoa não irá comer porque sabe que ele lhe causou problemas, mesmo que tenha apreciado muito o alimento. Supondo que esta pessoa estivesse passando fome e este alimento fosse a única comida encontrada.

Ele comeria para matar a fome e não morrer de inanição, mesmo correndo o risco da alergia. Estes exemplos ilustram uma prioridade de comportamentos e no primeiro caso o comportamento “evitar comer” teve prioridade sobre “comer” o alimento. Mas no segundo a prioridade do comportamento “sobreviva” sobrepôs o comportamento “não coma”.

Este tipo de prioridade é que dará base para implementação dos comportamentos no AA simulado, porque ele terá que fazer tarefas em que alguns comportamentos emergirão, contudo deverá assumir apenas um dos comportamentos. Isto ocorrerá quando ele tiver que, por exemplo, carregar suas baterias ou atingir um outro objetivo, exemplo citado na “seção motivação” no capítulo Introdução.

O AA encontrará situações que terá que superar obstáculos para atingir o seu objetivo ou desistir dele no caso de “correr perigo”. Se a intenção é obter AA’s com comportamentos e inteligência semelhantes a um animal, então, também terá que ter este tipo de controle.

Quando o Robô simulado entrar em situações que seja necessária a escolha do melhor comportamento, a rede neural hierárquica será a diferença entre implementações anteriores e esta, pois esta arquitetura favorecerá a escolha e priorização de um comportamento.

Imaginemos a seguinte situação, o robô está andando pelo mundo e detecta um foco luminoso, e deverá ir em sua direção até estar muito próximo. No caso do ambiente simulado o foco luminoso também é considerado como obstáculo, já que se trata de um “abajur”.

No caso de encontrar um obstáculo, quando estiver indo para o foco luminoso, que impeça a sua aproximação em relação ao foco, ele deverá desviar primeiro para depois ir de encontro ao foco. Então, seu comportamento deverá ser não se chocar com este obstáculo que impede a aproximação, e sim, tentar dar a volta e chegar ao objetivo.

Definiu-se para esta parte do trabalho que encontrar um foco luminoso é um comportamento que deve emergir logo que um foco for detectado, no entanto para isto acontecer o robô terá que desviar dos obstáculos. Os obstáculos formando paredes ou somente “tijolos” espalhados pelo mundo são translúcidos, possibilitando a detecção do foco de luz através deles.

O exemplo acima está sendo mostrado pelas figuras a seguir.

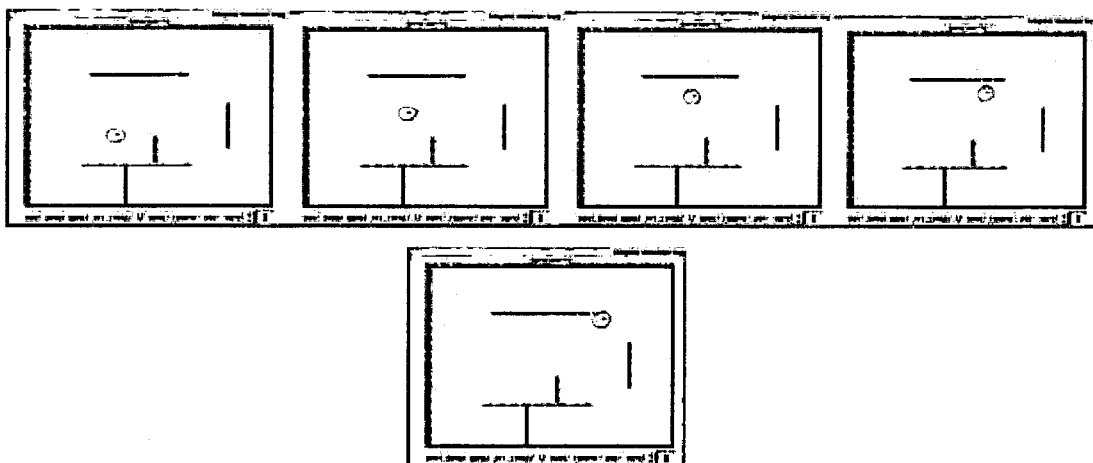


Figura 4.4: Prioridade de comportamento

As figuras mostram o Robô simulado detectando a presença de um foco de luz através de uma parede translúcida e contornando-a para chegar ao foco. Este comportamento terá que ocorrer e o robô não poderá se chocar.

“Para selecionar o programa de ação adequado para manter o animal em condições ótimas, o SNC tem que dispor de informação sobre os eventos no ambiente exterior. Estas informações são coletadas pelos órgãos dos sentidos ou receptores sensoriais. A resposta a um estímulo é inicialmente forte mas se enfraquece com o decorrer do tempo, uma característica denominada adaptação sensorial. Esta é uma propriedade biologicamente muito importante, pois auxilia o SNC a separar a informação relevante da irrelevante.” (MESSENGER,1980)

Podemos considerar todas as informações recebidas pelos sensores como relevantes, mas no caso do robô, a RNAH (Rede Neural Artificial Hierárquica) determinará qual informação é mais importante, acionando assim o comportamento que poderá manter o robô simulado em ótimas condições.

4.1.3 Descrição dos Comportamentos Implementados

Os comportamentos implementados resultaram em um controle eficaz para o Robô Khepera ou, numa visão ampla, o controle eficaz de vários tipos de AA's.

Buscamos deixar estes AA's com um alto grau de inteligência e autonomia. Muitos pesquisadores da área de IA conseguiram resultados considerados importantes para esta busca, usando Redes Neurais, Algoritmos Genéticos, entre outras. Uma opção é tentar melhorar as arquiteturas existentes ou criar novas, de maneira que fiquem mais eficazes.

Nossa proposta buscou atingir um grau maior de autonomia e inteligência, baseando-nos em algumas arquiteturas como, por exemplo, a proposta por BROOKS (1990b), chamada de “The Subsumption Architecture”, que está comentada no capítulo 3.

4.1.3.1 Rede Sensória de Distância e de Luminosidade

Temos que entender, em primeiro lugar, o funcionamento das redes do Subsistema Sensório. A primeira rede que veremos é a Rede Sensória de Distância, chamada KhepDist.

1. Quando a rede detectar qualquer obstáculo em qualquer um dos seus sensores, ela terá como saída o valor 1.
2. Quando a rede não detectar obstáculo, ela terá como saída o valor -1.
3. Esta rede ficará ativa enquanto o robô estiver “andando” pelo mundo.

A segunda rede é a Rede Sensória de Luz, chamada KhepLuz. De certa forma, ela age como a rede anteriormente vista.

1. Quando a rede detectar um foco de luz, sua saída será 1.
 2. Quando a rede não detectar o foco de luz, ela terá como saída o valor -1.
 3. Esta rede funcionará enquanto o robô estiver “andando” pelo mundo.
- Então, ela funcionará em paralelo com a rede de distância.

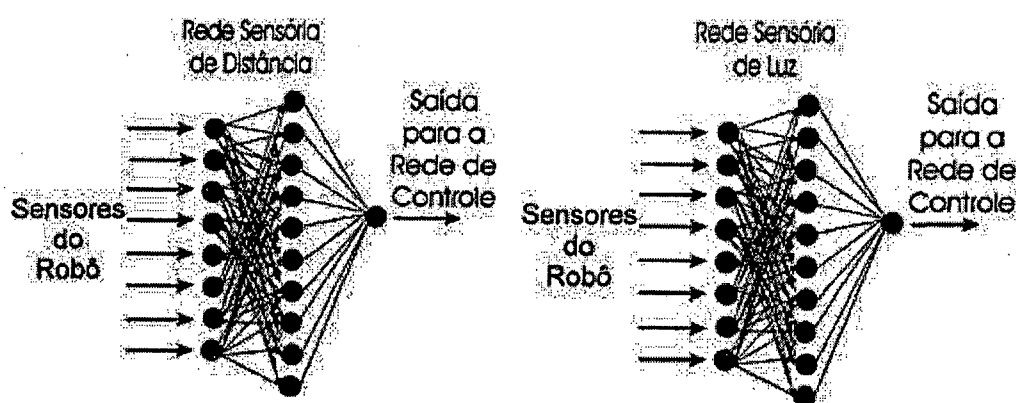


Figura 4.5: Redes Sensórias de Distância e de Luz.

Este Sistema Sensório, composto por estas duas redes, ficará responsável por indicar se o robô encontrou luz, obstáculo ou as duas coisas ao mesmo tempo. Digamos que, se fosse a descrição de um animal, este sistema seria os órgãos de sentido.

4.1.3.2 Rede de Controle dos Comportamentos Reflexivos e Reativos

Os valores de saída das redes do subsistema sensório fazem parte das entradas da rede de controle. A rede de controle é do tipo recorrente, então, as suas entradas serão as saídas das redes sensórias e a retroação. Esta parte da hierarquia tem o nome de Subsistema de Controle. Ela é responsável pela “memória” do robô e deverá organizar as suas ações.

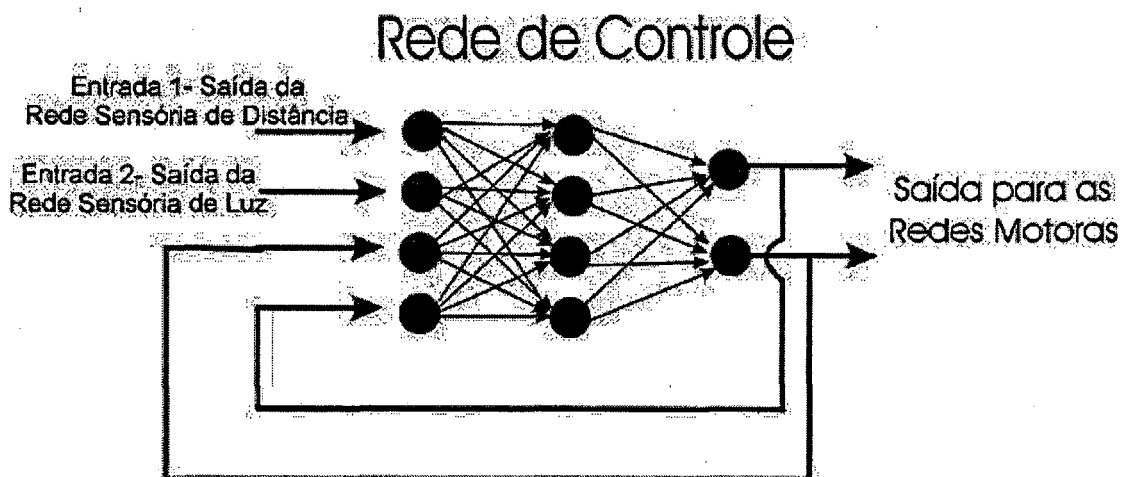


Figura 4.6: Rede de Controle – Hierarquia dos Comportamentos Reflexivos e Reativos

As ações dos motores não dependerão apenas do que é encontrado pela frente, isto é, se os sensores acusarem um foco de luz o robô poderá não ir em sua direção, pois dependerá também do estado em que se encontra a rede de controle.

Esta “atitude” deve-se ao fato de que se um foco luminoso for encontrado atrás de um obstáculo, o robô dirigirá-se para o foco podendo chocar-se com este obstáculo. Este tipo de comportamento é inaceitável, por isto devemos planejar muito bem o que o robô poderá ou não fazer. Os “comportamentos esperados” por esta rede serão descritos a seguir.

Em primeiro lugar, o robô circulará pelo mundo, recebendo valores nos seus sensores. Estes valores podem não indicar presença de obstáculo ou luz e isto fará com que as saídas das redes sensórias, KhepDist e KhepLuz, sejam respectivamente -1 e -1. Esta situação levaria o robô a ficar circulando a procura de luz.

Como já citado, ela representa um AEF com três estados. Estes estados indicam a situação na qual o robô se encontra, isto é, se ele não encontrar obstáculo e nem luz o estado será “Vagar/Desviar”, estado (0); se encontrar luz será “Encontrou Foco de Luz”, estado (1) e por último o estado (2) “Encontrou Obstáculo”. Como é mostrado na figura a seguir:

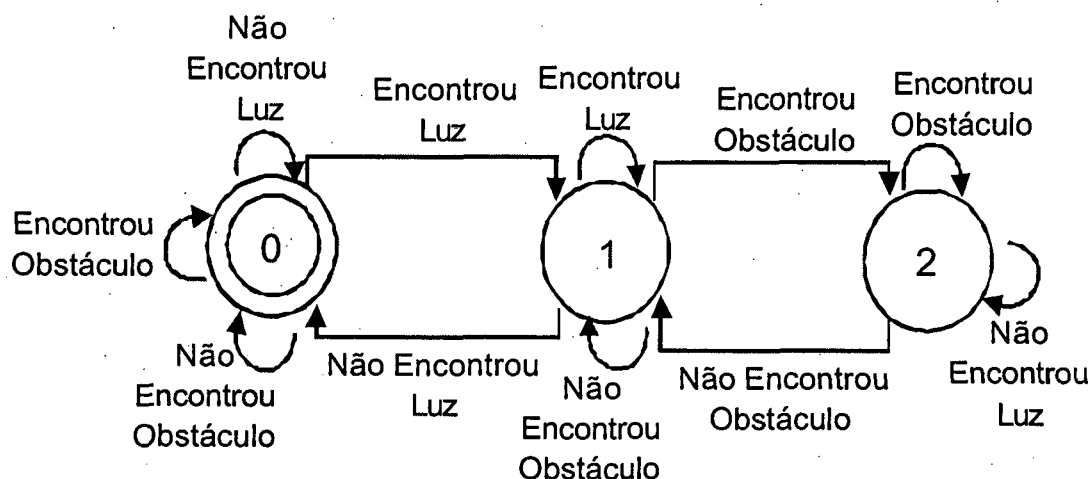


Figura 4.7: Representação da Rede de Controle Através de um AEF

As situações esperadas devido ao encontro ou não de obstáculos e luz são:

1. No caso da rede de controle estar no estado 0 (Vagar/Desviar)
 - Se o robô não encontrar obstáculo ou luz, ele vagará.
 - Se o robô encontrar obstáculo, ele irá se desviar.
 - Se o robô encontrar luz, a rede de controle passará para o estado 1 (Encontrou Luz) e ele irá de encontro à luz.
 - Se o robô encontrar luz atrás de um obstáculo, a rede de controle permanecerá no estado 0 e dará prioridade ao desvio deste obstáculo.
2. No caso da rede de controle estar no estado 1 (Encontrou Luz).
 - Se o robô encontrou luz, a rede permanecerá no estado 1 e ele irá de encontro à luz.
 - Se o robô encontrou obstáculo, ela passará para o estado 2 (Desviar de Obstáculo) e o robô desviará do obstáculo.

- Se o robô encontrou uma luz atrás de um obstáculo, a rede de controle passará para o estado 2 (Desviar de obstáculo) e o robô irá se desviar deste obstáculo.
 - Se o robô não detectar obstáculo ou luz, a rede de controle passará para o estado 0 (Vagar/Desviar) e o robô ficará vagando pelo mundo até encontrar algo.
3. No caso da rede de controle estar no estado 2 (Desviar de Obstáculo).
- Se o robô não encontrar nada a sua volta, a rede de controle passará para o estado 0 (Vagar/Desviar) e o robô irá vagar até encontrar algo.
 - Se o robô encontrar um foco de luz, a rede de controle passará para o estado 1 (encontrou luz) e o robô irá de encontro à luz.
 - Se o robô encontrar um foco de luz atrás de um obstáculo, a rede de controle permanecerá no estado 2 (Desviar de Obstáculo) e o robô irá priorizar o desvio deste obstáculo.
 - Se o robô encontrar um obstáculo, a rede de controle permanecerá no estado 2 (Desviar de Obstáculo) e o robô irá desviar do obstáculo.

4.1.3.3 Rede Motora de Distância e de Luminosidade

As duas próximas redes fazem parte do Subsistema Motor, sendo a Rede Motora de Distância, chamada KhepMotD e a Rede Motora Luz, chamada KhepMotL.

Estas duas redes serão responsáveis pela movimentação do robô no mundo, isto é, controlarão os motores. A configuração destas duas redes, na camada de entrada, é de 10 neurônios. Destes 10 neurônios, 2 são referentes as saídas da rede de controle.

O significado disto é, dependendo do estado que as saídas da rede de controle estiverem indicando, a rede será habilitada ou desabilitada, como veremos a seguir.

O primeiro caso a verificar será o da Rede Motora de Distância (KhepMotD):

- Se a rede sensória de distância (KhepDist) estiver detectando um obstáculo, ela passará a informação para a rede de controle. Por sua vez, a rede de controle irá indicar em qual estado ela está; este estado tem a propriedade de habilitar ou desabilitar a rede KhepMotD. Então, se os sensores da rede KhepMotD estiverem detectando um obstáculo, significará que a rede de controle estará no estado 0 (Vagar/Desviar) ou no estado 2 (Desviar de Obstáculo). Isto ocasionará a habilitação desta rede e a inibição da Rede Motora Luz, mesmo que esta indique a presença de um foco de luz. A ação nos motores será por conta da rede KhepMotD, indicando para onde o robô deve dirigir-se e provocando o desvio do obstáculo.

O segundo caso que iremos verificar é o da Rede Motora Luz (KhepMotL):

- No caso da rede sensória de luz (KhepLuz) detectar um foco de luz e a rede KhepDist não detectar obstáculo, aquela passará a informação para a rede de controle. Isto fará com que a rede de controle, independente do estado que esteja, vá para o estado 1 (Encontrou Luz), habilitando, assim, a rede KhepMotL e desabilitando a rede KhepMotD. Isto implicará a busca pelo foco de luz, pois quem controlará os motores é a rede KhepMotL, através dos seus sensores que indicarão onde o foco está.

Para haver a atuação sobre os motores, é necessária a soma de cada saída da rede KhepMotD com sua respectiva saída na rede KhepMotL. Então será somada a saída referente ao Motor Esquerdo da rede KhepMotD com a saída referente ao Motor Esquerdo da rede KhepMotL, equação (1), para ter o valor que será aplicado ao motor esquerdo. Isto também será feito para o Motor Direito, equação (2).

$$\text{Robot->Motor[LEFT].Value} = (\text{SaidaMotL1} + \text{SaidaMotD1}) * 5; \quad (1)$$

$$\text{Robot->Motor[RIGHT].Value} = (\text{SaidaMotL2} + \text{SaidaMotD2}) * 5; \quad (2)$$

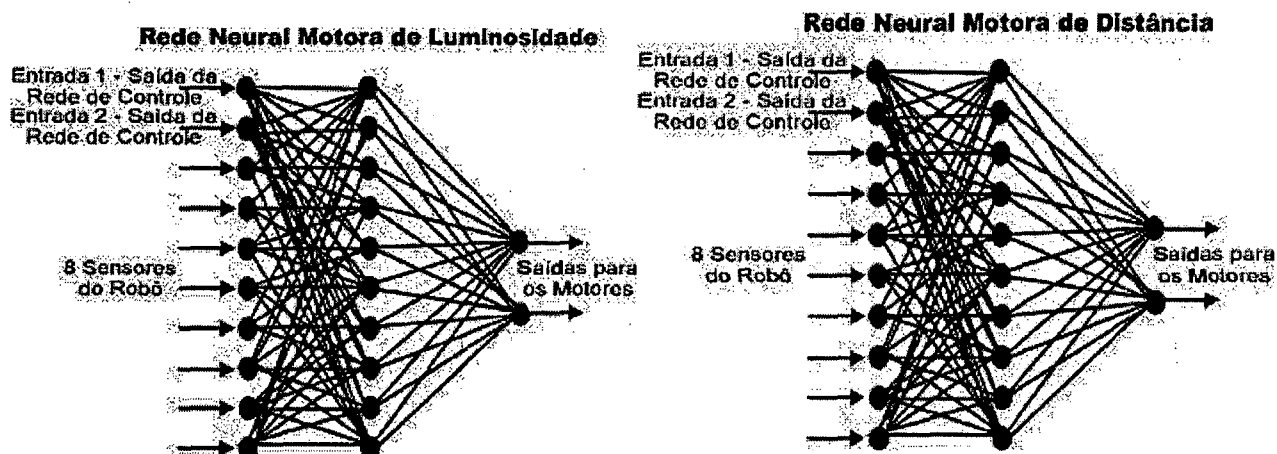


Figura 4.8: Redes Motoras de Distância e de Luz

O resultado da soma entre os valores das saídas que variam de -1 até 1 são normalizados para valores que o simulador entenda, por isto, é multiplicado por 5, que é o valor médio de aceleração do robô, para que ele não fique muito rápido, impossibilitando visualmente a coleta de informações pelos usuários. Esta hierarquia foi comentada em quase todos os capítulos e abaixo segue a figura mostrando como ela foi idealizada:

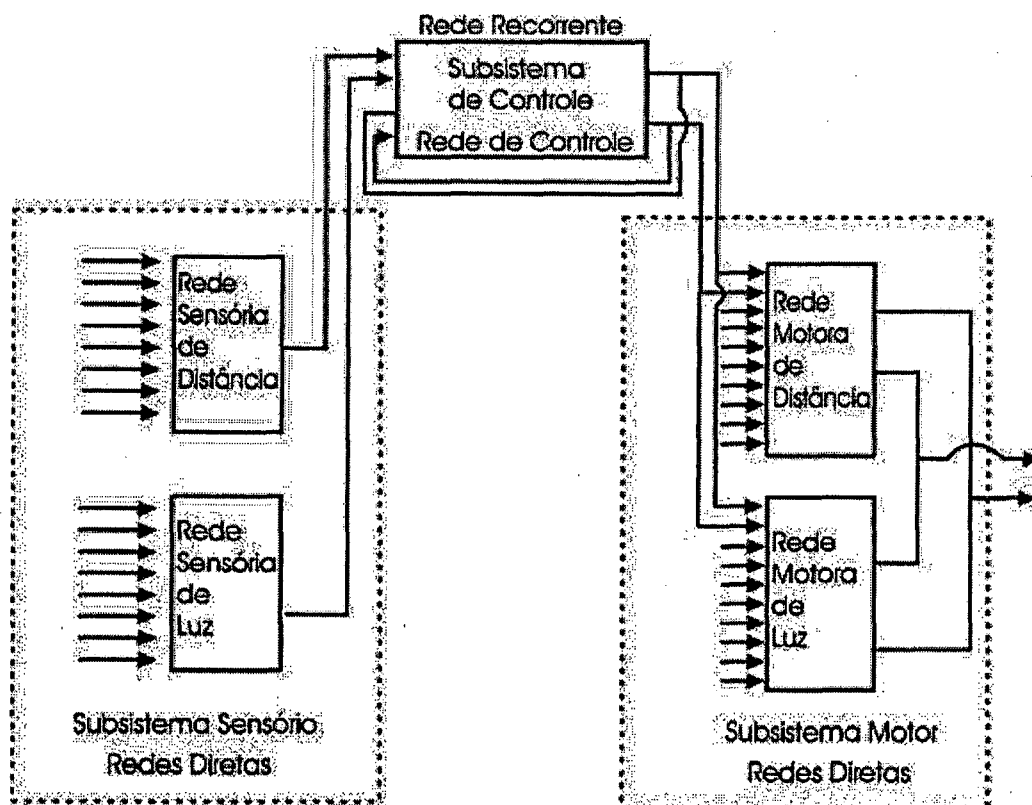


Figura 4.9: Representação Gráfica da Rede Neural Hierárquica.

4.2 Comportamentos Reativos, Reflexivos e Controle de Energia

4.2.1 Alterações na Estrutura da Hierarquia

Como a arquitetura anterior gerou resultados satisfatórios, que serão vistos no capítulo 5, passamos para a etapa seguinte que é a criação de um comportamento diferente dos já implementados. Esta implementação acarretará uma alteração na hierarquia das redes.

As redes, KhepDist e KhepLuz, do Subsistema Sensorio, serão mantidas com a mesma estrutura e com os mesmos conjuntos de entradas e saídas. Um novo comportamento reflexivo será implementado e alterações serão realizadas no Subsistema de Controle e no Subsistema Motor.

As alterações implementadas visam aumentar a complexidade dos comportamentos do robô, isto é, aumentando o número de comportamentos reativos e reflexivos, automaticamente, aumentamos a complexidade do AA.

Os comportamentos terão que trabalhar em conjunto e a falha de um poderá comprometer o conjunto, o que não ocorria com a “Subsumption Architecture” proposta por BROOKS (1990b).

Este comprometimento pode ser explicado, por exemplo, pela falha da rede sensoria de obstáculo que, muito provavelmente, provocaria a colisão do robô com os obstáculos.

Uma vez que a rede sensoria de obstáculo não manda a informação para a rede de controle, esta não terá como saber quando o robô está próximo dos obstáculos. Então, a falha é caracterizada pela colisão com o obstáculo, um evento não permitido para a “integridade física” do robô.

A nova configuração da hierarquia passa a ter 7 (sete) redes neurais e um perceptron de controle de energia. A descrição de toda a hierarquia será por nível, isto significa que será descrito primeiramente o nível mais baixo, as redes diretas sensórias e motoras.

Em seguida o nível intermediário onde estão as redes recorrentes responsáveis pelos comportamentos reativos. Por fim, o nível mais alto responsável pelo controle de energia do robô. A figura abaixo mostra a representação gráfica da nova arquitetura.

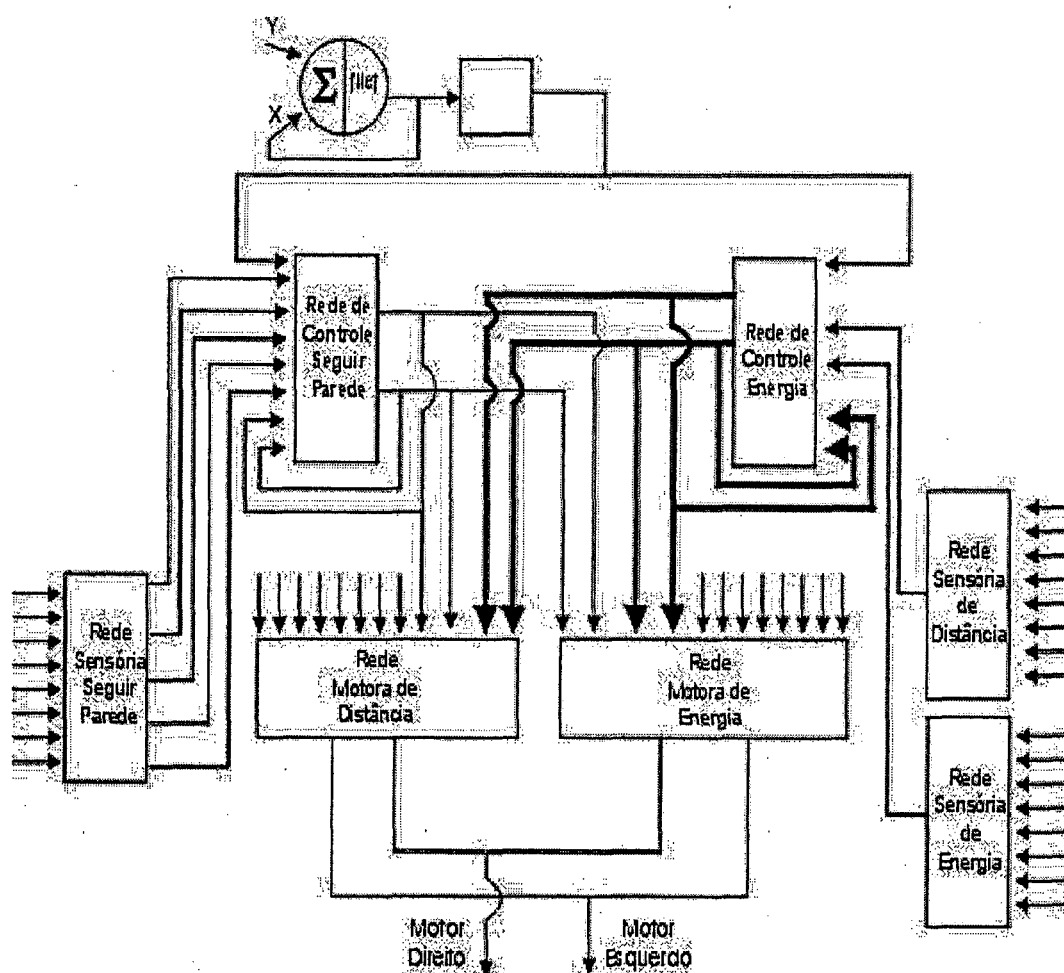


Figura 4.10: Representação Gráfica da Nova Rede Neural Hierárquica.

O algoritmo básico, a seguir, procura dar um auxílio no entendimento da hierarquia, porém os níveis serão detalhados na seção “Descrição dos Novos Comportamentos Implementados”, o que facilitará o entendimento de toda a estrutura.

➤ Algoritmo:

- Detectar Obstáculos (Rede Sensória de Distância – Subsistema Seguir Parede)
- Detectar Obstáculos (Rede Sensória de Distância – Subsistema Controle de Energia)
- Detectar Energia (Rede Sensória de Energia – Subsistema Controle de Energia)
- Se Energia = Normal (Perceptron)
 - Ativar Rede de Controle Seguir Parede (Rede Controle Seguir Parede – Subsistema Seguir Parede)
- Senão
 - Ativar Rede de Controle Buscar Energia (Rede Controle de Energia – Subsistema Controle de Energia)
- Se Rede de Controle Seguir Parede Ativada
 - Ativar Rede Motora Distância (Rede Motora de Distância – Subsistema Motor)
- Se Rede de Controle Buscar Energia Ativada
 - Se encontrou Obstáculo
 - Ativar Rede Motora Distância (Rede Motora de Distância – Subsistema Motor)
 - Se encontrou Energia
 - Ativar Rede Motora Energia (Rede Motora de Energia – Subsistema Motor)

Basicamente, a função da hierarquia é detectar obstáculos e/ou foco de energia, e passar a informação para as redes de controle. A rede de controle, que estará ativa, depende do controle de energia, isto significa que uma estará ativa quando a energia estiver normal e a outra, quando a energia estiver baixa. As redes motoras irão atender ao comando das redes de controle, cada qual com sua função. A Rede de Distância irá controlar o comportamento de seguir parede e também de desviar de obstáculos, e a rede de Energia irá procurar um foco de energia e quando achar irá ao seu encontro.

4.2.2 Descrição dos Novos Comportamentos Implementados

Como citado acima, a primeira rede que será descrita é a Rede Sensória de Distância, uma rede direta, a qual informará a existência ou não de “parede” e “obstáculos”.

A definição de parede e obstáculo confunde-se, pois tudo o que é “físico”, diferente de um foco luminoso, pode e será considerado como parede, pelo fato de ser detectado pelos sensores de distância. Para exemplificar a definição acima, analisemos a seguinte situação hipotética:

Uma barata está em um canto de uma sala e quer ir para o outro, seguindo a parede, e no meio do caminho depara-se com um obstáculo encostado à parede. Ela terá que contornar o obstáculo para atingir o objetivo. Então, para ela o obstáculo era algo encostado à parede. Imagine agora que ela está indo de um canto para o outro, porém passando pelo meio da sala, e durante o percurso encontra um objeto grande, por exemplo, um sofá. Como a barata irá definir o que é “parede” e o que é “obstáculo”? Nesse caso a barata poderá, não é certeza, ficar dando voltas ao redor do sofá. Então, o que era obstáculo passou a ser “entendido” como uma parede. O robô simulado também terá o mesmo comportamento descrito acima, até que sua energia fique baixa.

4.2.2.1 Rede Sensória Seguir Parede

A Rede Sensória de Distância, do Subsistema Sensorio Seguir Parede, tem na sua configuração:

- 8 Neurônios na Camada de Entrada;
- 10 Neurônios na Camada Intermediária (oculta);
- 5 Neurônios na Camada de Saída.

Os neurônios da camada de entrada recebem os valores dos sensores de distância do robô simulado. Os valores dos sensores de distância variam de 0 (ausência de

obstáculos) até 1023 (choque com o obstáculo detectado). As saídas indicarão a ausência de obstáculos, e no caso de presença, a posição dos mesmos.

As saídas são nomeadas a seguir:

- Saída 1 - Presença de obstáculo na Parte Frontal: OF
- Saída 2 - Presença de obstáculo na Parte Esquerda: OE
- Saída 3 - Ausência de obstáculo na Parte Frontal: NOF
- Saída 4 - Ausência de obstáculo na Parte Esquerda: NOE
- Saída 5 - Ausência Total de Obstáculo: NO.

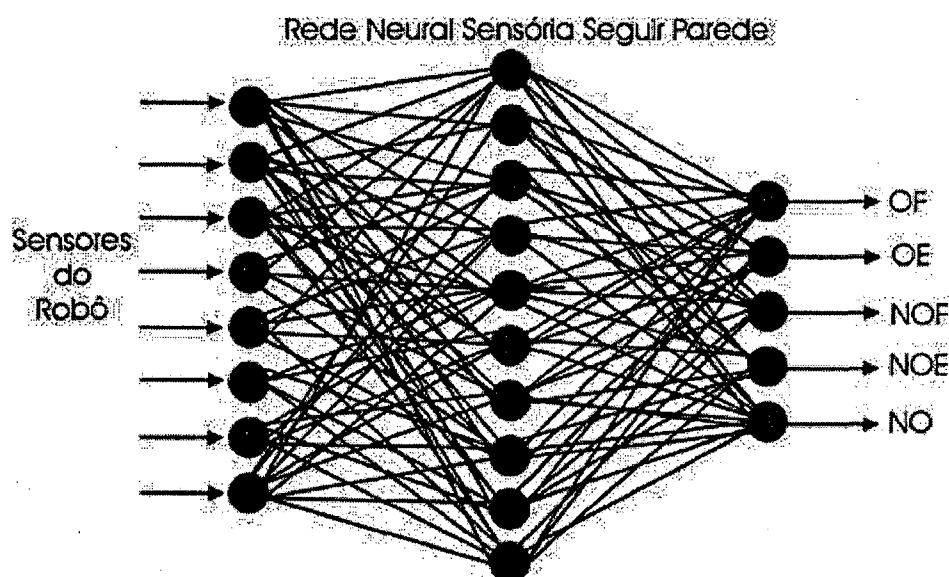


Figura 4.11: Rede Neural Direta Sensória – Seguir Parede

4.2.2.2 Rede Sensória de Distância e de Energia

A função da Rede Sensória de Distância, que tem sua saída fazendo parte das entradas da Rede de Controle de Energia, é a de informar a presença ou ausência de obstáculos, já que suas entradas são os 8 sensores de distância do robô.

A Rede Sensória de Energia tem a função de informar a presença ou ausência de um ponto de energia, uma vez que suas entradas são os 8 sensores de energia (luminosidade) do robô, e a sua saída também faz parte das entradas da Rede de Controle de Energia. As redes não se importam com a posição dos obstáculos ou ponto

de energia, apenas se existe ou não, sendo assim, elas trabalham como “condutoras de estímulos”, se é que podemos fazer uma comparação com as redes neurais biológicas. As redes acima conduzem o “estímulo” até a rede de controle, onde ele é identificado (obstáculo ou ponto de energia).

O funcionamento obedece regras que determinam qual “estímulo” deve ser enviado para a rede de controle, isto é, quando os sensores do robô (entradas da rede) detectam algo, o valor da saída será 1 e quando não detectam, o valor será -1. A rede de energia funciona da mesma forma. As configurações das duas são idênticas quanto ao número de neurônios nas camadas de entrada, intermediária (oculta) e saída.

Configuração das Redes Sensórias de Distância e de Energia:

- 8 Neurônios na Camada de Entrada;
- 10 Neurônios na Camada Intermediária (oculta);
- 1 Neurônio na Camada de Saída.

Rede Neural Sensória de Distância

Rede Neural Sensória de Energia

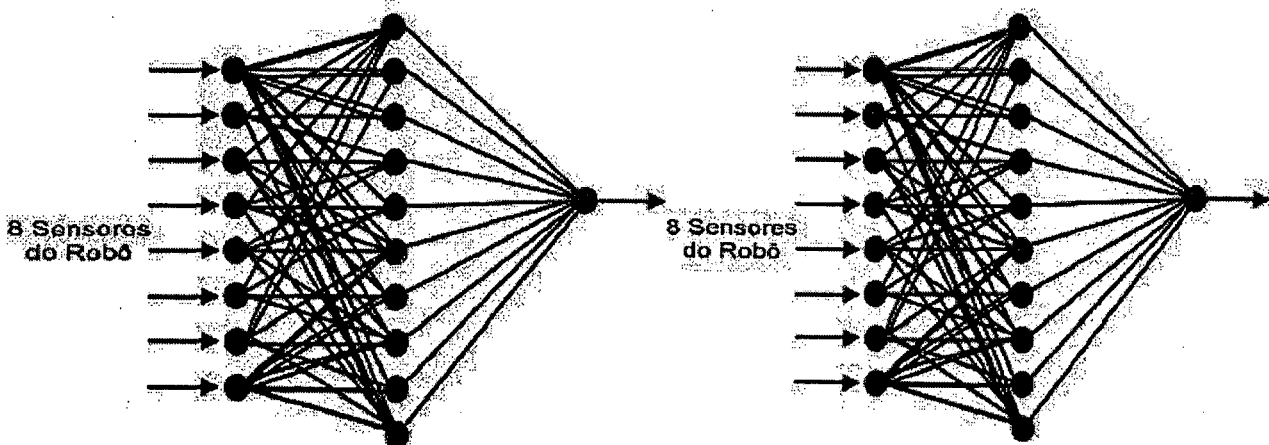


Figura 4.12: Redes Neurais Diretas – Sensória de Distância e de Energia

4.2.2.3 Controle de Energia

O controle de Energia pode ser comparado ao comportamento instintivo animal, já que no animal esse comportamento aparece, geralmente, quando ele necessita recarregar as suas energias. No robô simulado não temos a implementação da bateria do robô real, isto proporcionaria o controle do tempo que o robô simulado poderia “andar” pelo

mundo. Para simular esta autonomia, implementou-se um perceptron com uma equação de reta, equação 3, como função da sua saída.

$$F_{net} := 0.999912 * (X + Y) - 1,0000045 \quad (3)$$

Na equação, o valor de X é constante e corresponde a 1. O valor inicial de Y é 1 e após a primeira execução, passa a receber o valor calculado pela equação (F_{net}), o que significa uma realimentação. A saída irá decrescer de 0.9 (Energia Normal) até 0.3 (Energia Baixa). Estando no nível mais alto da hierarquia implementada, o controle de energia irá atuar sobre as duas redes de controle (Seguir Parede e Encontrar Ponto de Energia), ativando-as ou desativando-as através de um comando de seleção (if). Os valores de saída da equação entre 0.9000 e 0.3000 resultam na seleção do valor 1, pelo comando “if”, e os valores abaixo de 0.3000 resultam no valor 0. A figura abaixo mostra o gráfico da equação.

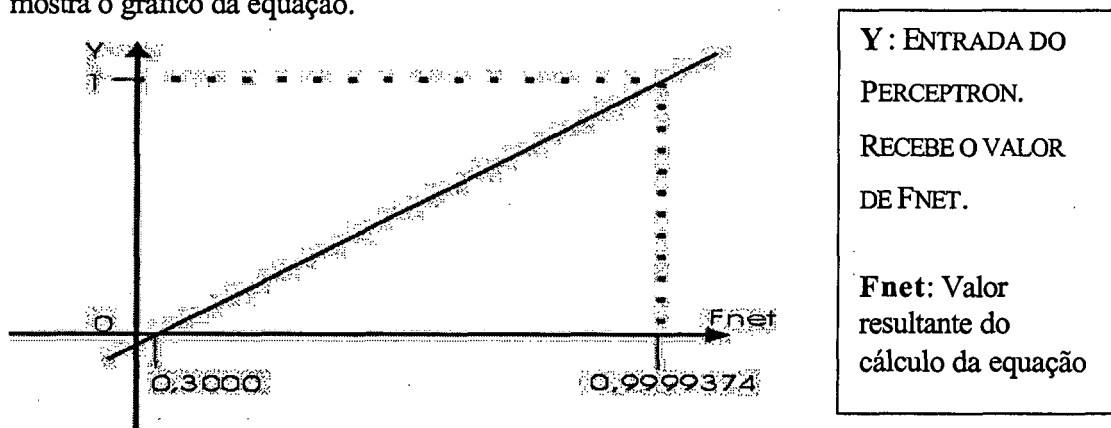


Figura 4.13: Gráfico da Função de Saída do Controle de Energia

O algoritmo do comando “IF”, para a seleção dos valores que farão parte das entradas das redes de controle, é mostrado a seguir:

- Se valor de $F_{net} > 0.3000$ então
 - Energia = 1 (Normal)
- Senão
 - Energia = 0 (Baixa)

A Figura a seguir mostra a representação do Controle de Energia através de um Autômato de Estado Finito.

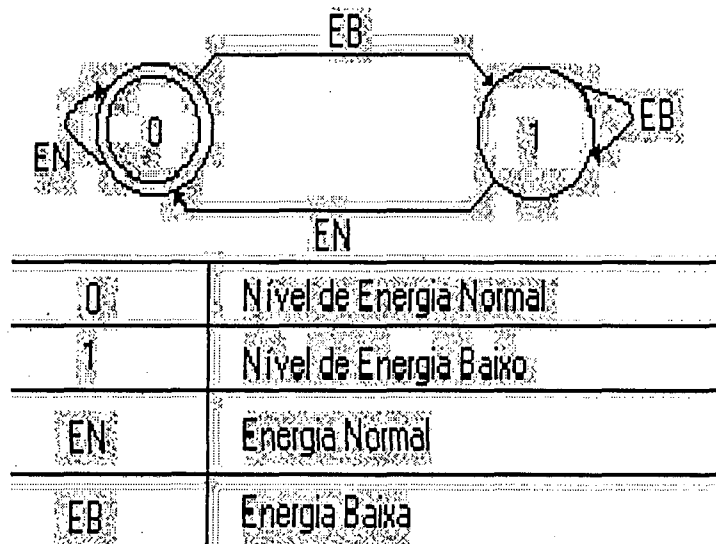


Figura 4.14: Autômato de Estado Finito do Controle de Energia

4.2.2.4 Rede de Controle Seguir Parede

A função da Rede de Controle Seguir Parede é a de manter o robô seguindo uma parede enquanto a energia dele estiver normal. Para que o robô siga a parede, a rede de controle recebe as informações da rede sensória seguir parede, indicando que ele encontrou um obstáculo ou parede.

De fato nem tudo que é detectado pelos sensores do robô, na rede sensória seguir parede, é considerado parede. Para ser considerada parede, os sensores da esquerda deverão estar ativos (detectando algo), caso contrário, o que está sendo detectado passa a ser obstáculo. Essa regra é apenas uma forma de validar a hierarquia e não uma regra adotada como única. O controle feito pela rede de controle é executado sobre as redes motoras, parte do seguinte princípio:

- Se os sensores do robô não detectarem nada, o estado da rede de controle será 0 (zero).
- Se os sensores frontais detectarem algo, o estado da rede de controle será 1 (um).
- Se os sensores da esquerda detectarem algo, o estado da rede será 2 (dois)

- parede à esquerda.
- Se os sensores da direita e/ou da traseira detectarem algo, o estado será o 3 (três) - nenhum “obstáculo” à direita e nenhum “obstáculo” na traseira.

A rede de controle representa um Autômato de Estado Finito (AEF) com 4 (quatro) estados, variando de 0 a 3, sendo 0 (zero) o estado inicial. O robô ficará no estado 0 (zero) enquanto os sensores não estiverem ativos (saída ajustada em -1). Para que a rede de controle se mantenha nesse estado, a rede sensória terá que estar com sua saída ajustada em NO (Nenhum objeto detectado).

A mudança de estado se dá quando um ou mais sensores, da rede sensória, estiverem ativos (saída ajustada em 1). Caso um ou os dois sensores frontais estiverem ativos (igual a 1), a rede sensória “informa” que o obstáculo está a frente (OF – Obstáculo Frontal), e o estado da rede de controle passa a ser o estado 1 (um).

Para que seja o estado 2 (dois), os sensores da esquerda deverão estar ativos, isso significa estar setado em 1 (detectou parede). Existirá caso em que os sensores da direita e/ou traseiros detectam o chamado obstáculo, tendo como consequência para a rede de controle a mudança para o estado 3 (três), significando nenhum obstáculo à esquerda e nem frontal (NOE – Nenhum Obstáculo à Esquerda e NOF – Nenhum Obstáculo Frontal).

Tem-se na configuração da Rede de Controle Seguir Parede:

- 8 Neurônios na Camada de Entrada;
- 10 Neurônios na Camada Intermediária (oculta);
- 2 Neurônios na Camada de Saída.

Das oito entradas, 1 (uma) é referente a saída do comando de seleção (IF), 5 (cinco) são as saídas da Rede Sensória Seguir Parede e as 2 (duas) restantes são a realimentação da Rede de Controle Seguir Parede. As saídas referem-se ao estado do AEF.

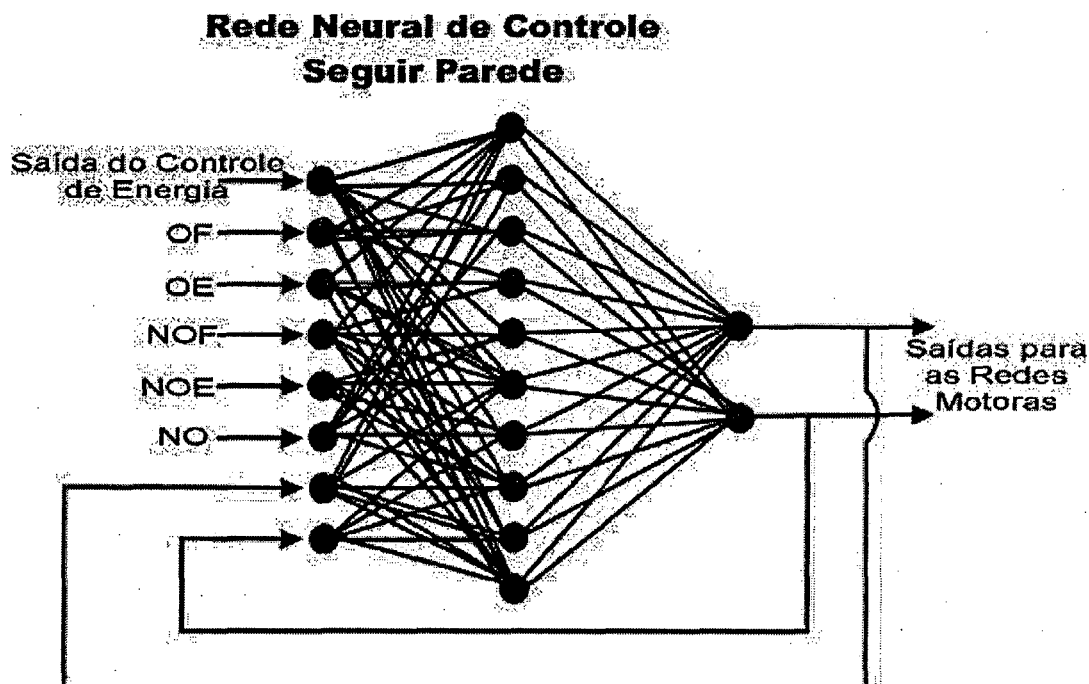


Figura 4.15: Rede Neural Recorrente de Controle – Seguir Parede

Abaixo segue o autômato representado pela rede descrita:

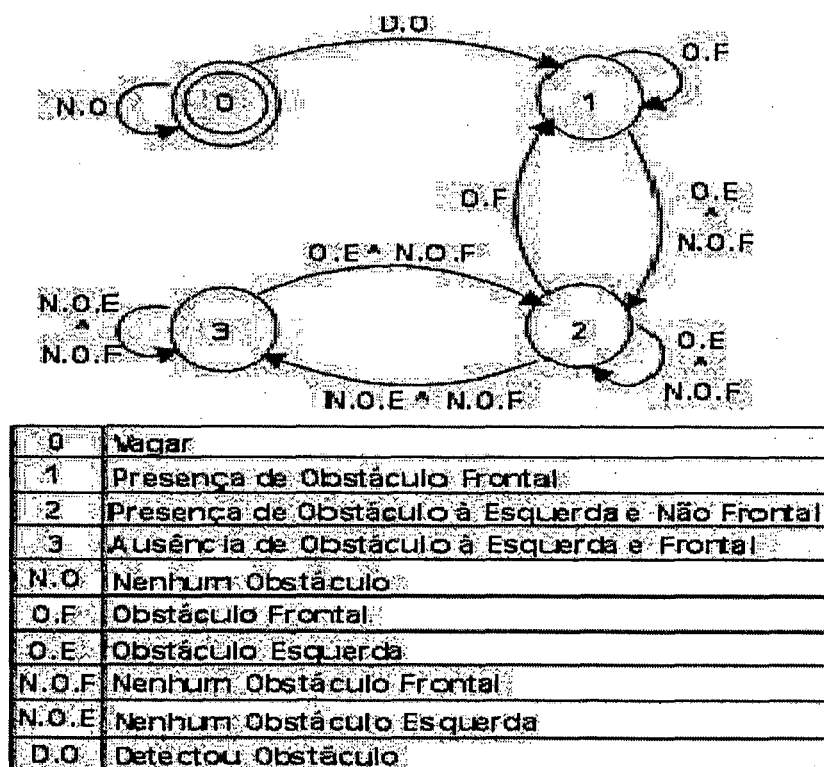


Figura 4.16: Autômato de Estado Finito (AEF) da Rede de Controle Seguir Parede.

4.2.2.5 Rede de Controle de Energia

A Rede de Controle de Energia passará a atuar sobre as redes motoras quando a energia do robô estiver baixa. Ela receberá as saídas das Redes Sensórias de Distância e Energia, possibilitando o controle das redes motoras.

Para encontrar o ponto de energia, a Rede de Controle de Energia deverá atuar de duas formas diferentes: o robô deverá desviar de qualquer coisa que seja detectado pelos sensores de distância e ir de encontro ao “Ponto de Energia”, quando esse for detectado pelos sensores de energia. Seu funcionamento é muito parecido com aquele implementado nos primeiros testes da hierarquia, mostrado na seção 4.1.3.3, página 58.

Estando a rede de energia no estado de energia baixa, fará com que o comando de seleção (IF) tenha sua saída em 0 (zero), habilitando a Rede de Controle de Energia e desabilitando a Rede de Controle Seguir Parede.

Nesse momento, o robô passará a percorrer o mundo a procura de um Ponto de Energia para poder reabastecer-se. Portanto, se durante o percurso deparar-se com algum obstáculo, a rede de controle habilita a rede motora de distância para o robô desviar do obstáculo.

Tudo o que é detectado pelos sensores de distância é considerado obstáculo. Ao detectar um ponto de energia o robô irá em sua direção para reabastecer-se, porém, encontrando um obstáculo que o impeça de atingir seu objetivo, a rede de controle desabilita a rede motora de energia e habilita a rede motora de distância, para essa desviar o robô do obstáculo.

Outra possibilidade é o robô estar com a energia baixa e ir de encontro a um ponto de energia sem encontrar obstáculos pela frente. A seguinte situação fará com que o robô chegue próximo ao ponto de energia e pare seus motores para simular o reabastecimento.

Ele ficará inerte por alguns segundos e depois voltará a percorrer o mundo procurando uma parede, pois estará sendo controlado pela Rede de Controle Seguir Parede, até que sua energia volte a ficar baixa.

Todos os passos que a Rede de Controle de Energia pode assumir são:

- A Rede Sensória de Distância detectou um obstáculo, sua saída será 1 (um) e o estado da Rede de Controle é 0 (zero). A Rede de Controle permanecerá no estado 0 (zero).
- A Rede Sensória de Distância detectou um obstáculo, sua saída será 1 (um) e o estado da Rede de Controle é 1 (um). A Rede de Controle passará para o estado 2 (dois).
- A Rede Sensória de Distância detectou um obstáculo, sua saída será 1 (um) e o estado da Rede de Controle é 2 (dois). A Rede de Controle permanecerá no estado 2 (dois).
- A Rede Sensória de Energia detectou um Ponto de Energia, sua saída será 1 (um) e o estado da Rede de Controle é 0 (zero). A Rede de Controle passará para o estado 1 (um).
- A Rede Sensória de Energia detectou um Ponto de Energia, sua saída será 1 (um) e o estado da Rede de Controle é 0 (zero). A Rede de Controle passará para o estado 1 (um).
- A Rede Sensória de Energia detectou um Ponto de Energia, sua saída será 1 (um) e o estado da Rede de Controle é 1 (um). A Rede de Controle permanecerá no estado 1 (um).
- A Rede Sensória de Energia detectou um Ponto de Energia, sua saída será 1 (um) e o estado da Rede de Controle é 0 (zero). A Rede de Controle passará para o estado 1 (um).
- A Rede Sensória de Energia detectou um Ponto de Energia, então sua saída será 1 (um) e o estado da Rede de Controle é 2 (dois). A Rede de Controle passará para o estado 1 (um).

- A Rede Sensória de Energia detectou um Ponto de Energia e a Rede Sensória de Distância detectou um obstáculo, as saídas de cada uma será 1 (um). Isso provocará uma prioridade de comportamento, no caso, será o desviar de obstáculos.
- A Rede Sensória de Energia e a Rede Sensória de Distância não detectaram nada, as saídas de cada uma será -1 (um negativo). Isso provocará uma prioridade de comportamento, no caso, será o vagar/desviar de obstáculos (estado 0 “zero” da Rede de Controle).

Pode-se perceber, nos casos em que a Rede Sensória de Distância está com sua saída em 1 (um), a Rede Sensória de Energia poderá estar com sua saída também em 1 (um) ou em -1 (um negativo) e vice-versa. E poderá haver casos em que as duas estarão com suas saídas em 1 (um) ou em -1 (um negativo).

Um fato que ocorre sempre é a prioridade de determinado comportamento sobre outro, mas a prioridade, sempre que a Rede Sensória de Distância estiver com sua saída em 1 (um), será do comportamento desviar de obstáculos.

A Representação da Rede de Controle de Energia, através de um AEF, é mostrada a seguir:

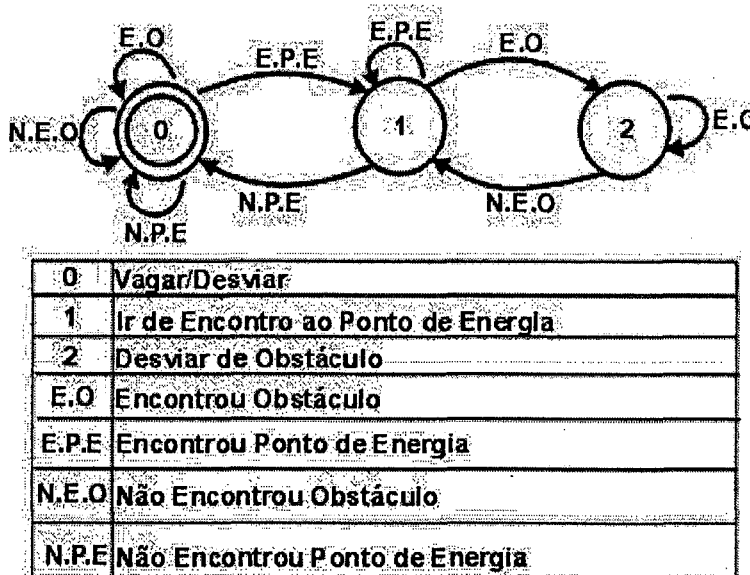


Figura 4.17: Autômato de Estado Finito (AEF) da Rede de Controle de Energia.

Então, temos na configuração dessa rede:

- 5 Neurônios na Camada de Entrada;
- 10 Neurônios na Camada Intermediária (oculta);
- 2 Neurônios na Camada de Saída.

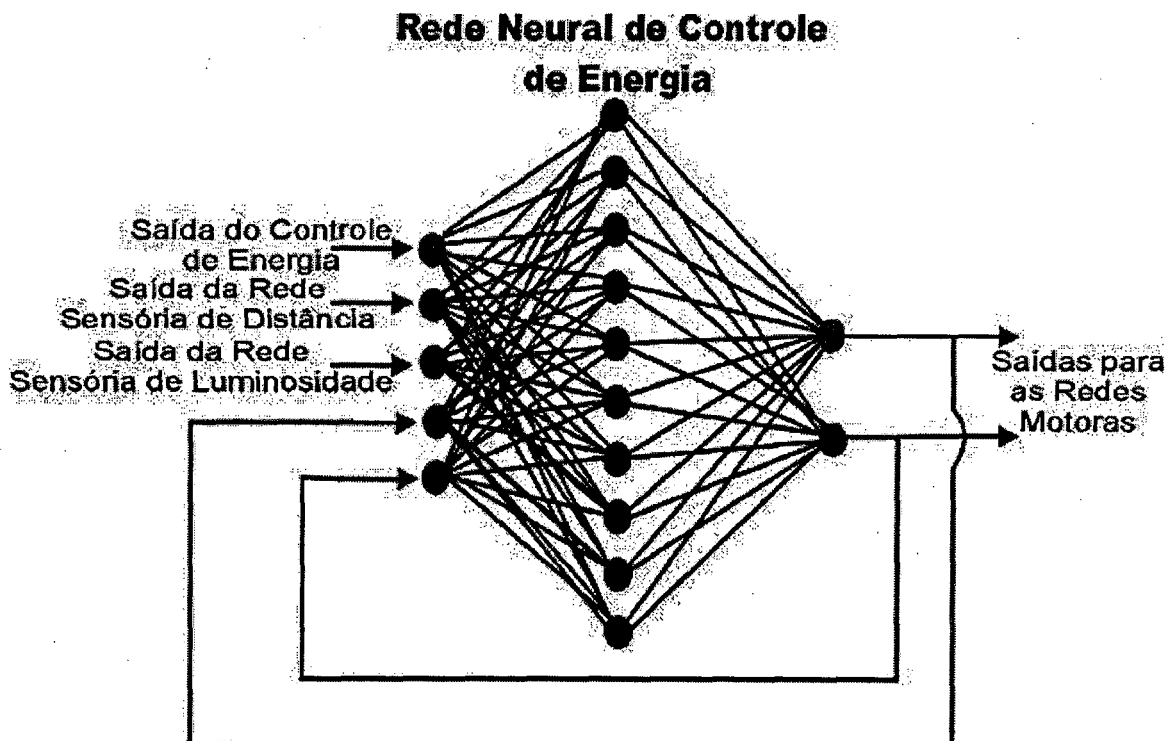


Figura 4.18: Rede Neural Recorrente de Controle - Energia

4.2.2.6 Rede Motora de Distância da Nova Arquitetura

Todo o movimento do robô depende das Redes Motoras, uma vez que as Redes de Controle fazem o papel de “Coordenadoras” dos comportamentos. O maior trabalho recai sobre a Rede Motora de Distância a qual sofre atuação de duas formas:

- Atuação da Rede de Controle Seguir Parede.
- Atuação da Rede de Controle de Energia.

A primeira atua fazendo o robô seguir parede e a segunda fazendo o robô desviar

de qualquer coisa detectada pelos sensores de distância. Esse tipo de controle não foi criado ao acaso, pois a configuração da hierarquia levou a ter apenas uma rede fazendo o “controle reflexivo”, baseado nos sensores de distância. Podemos comparar as duas atuações com alguns comportamentos, por exemplo, de humanos, já que os comportamentos do robô tiveram sua base na inspiração biológica. Imagine a seguinte situação em um humano: o andar - ato executado através de um comportamento instintivo. Golpe de uma arte marcial com os pés – comportamento desenvolvido através de aprendizado. E o Reflexo Patelar executado através de um golpe abaixo da rótula do joelho - comportamento reflexivo.

Estes três comportamentos são executados pelos músculos da perna e poderíamos generalizar dizendo que a perna é em última instância, acionada pelos motoneurônios de Rede Neural. Isto nos levaria a dizer que os três comportamentos agem em cima de uma mesma rede neural, porém, sofrendo a ação de três diferentes controles. As entradas dessa rede contêm os sensores do robô e as saídas das redes de controle, com isso sua configuração ficou da seguinte forma:

- 12 Neurônios na Camada de Entrada
- 14 Neurônios na Camada Intermediária (oculta)
- 2 Neurônios na Camada de Saída

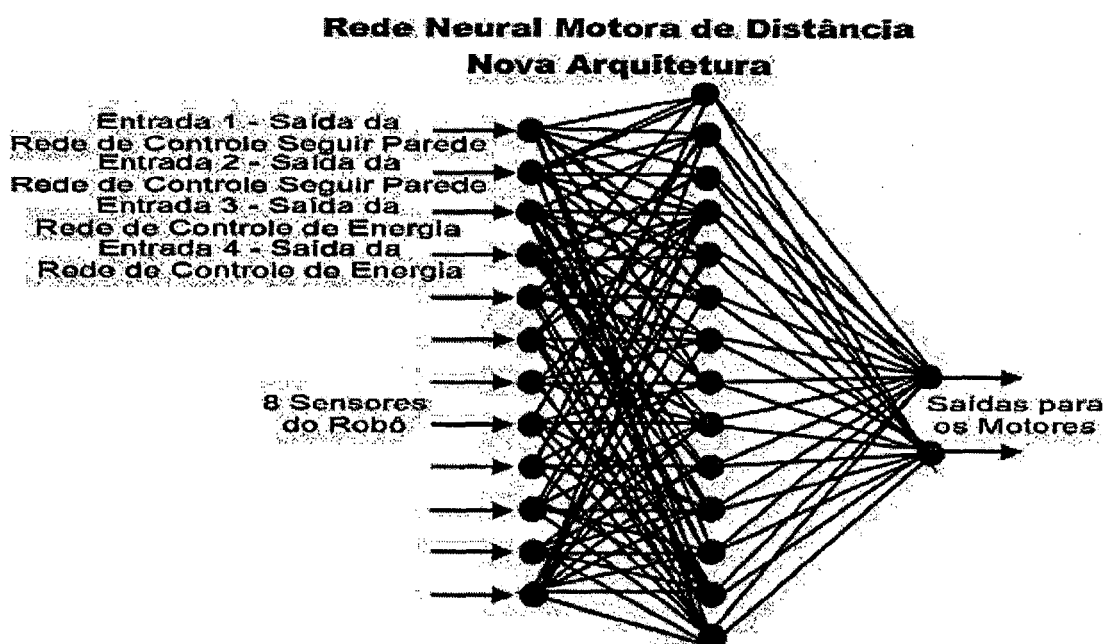


Figura 4.19: Rede Neural Direta Motora – Distância

4.2.2.7 Rede Motora de Energia

Essa rede atuará quando o robô detectar um ponto de energia e não encontrar obstáculo pela frente. Sua atuação depende de um espaço livre entre o momento em que detectou o ponto e seu trajeto até ele. No caso de encontrar um obstáculo, primeiro deverá desviar dele, para depois ir novamente ao encontro do ponto, e isto depende da rede de controle.

Quando um ponto de energia é encontrado e a energia do robô está baixa, a rede de controle passa para o estado 2 (dois), Ir de encontro ao ponto de energia, e envia a “mensagem” para a Rede Motora de Energia.

Sendo os sensores de energia entradas para a Rede Motora de Energia, ir de encontro ao ponto dependerá de qual sensor estará ativo. Estando, por exemplo, os sensores traseiros ativos, o robô irá girar no seu eixo, para o ponto de energia ser detectado pelos sensores frontais, requisito principal para ir de encontro ao ponto de energia e se posicionar para o reabastecimento.

Não importa qual sensor detecte o ponto de energia, pois o robô sempre deverá chegar de frente com o ponto, ou seja, os sensores frontais terão que estar detectando o ponto.

Assim que o robô alcançar uma distância, a qual seja o suficiente para o reabastecimento e para não colidir com o ponto de energia, seus motores serão desligados e ele ficará inerte por alguns segundos. Após o reabastecimento, o controle passa para a Rede de Controle Seguir Parede e a Rede Motora Seguir Parede.

A configuração dessa rede é idêntica, com relação ao número de neurônios, à Rede Motora Seguir Parede, como é mostrado abaixo:

- 12 Neurônios na Camada de Entrada
- 14 Neurônios na Camada Intermediária (oculta)
- 2 Neurônios na Camada de Saída

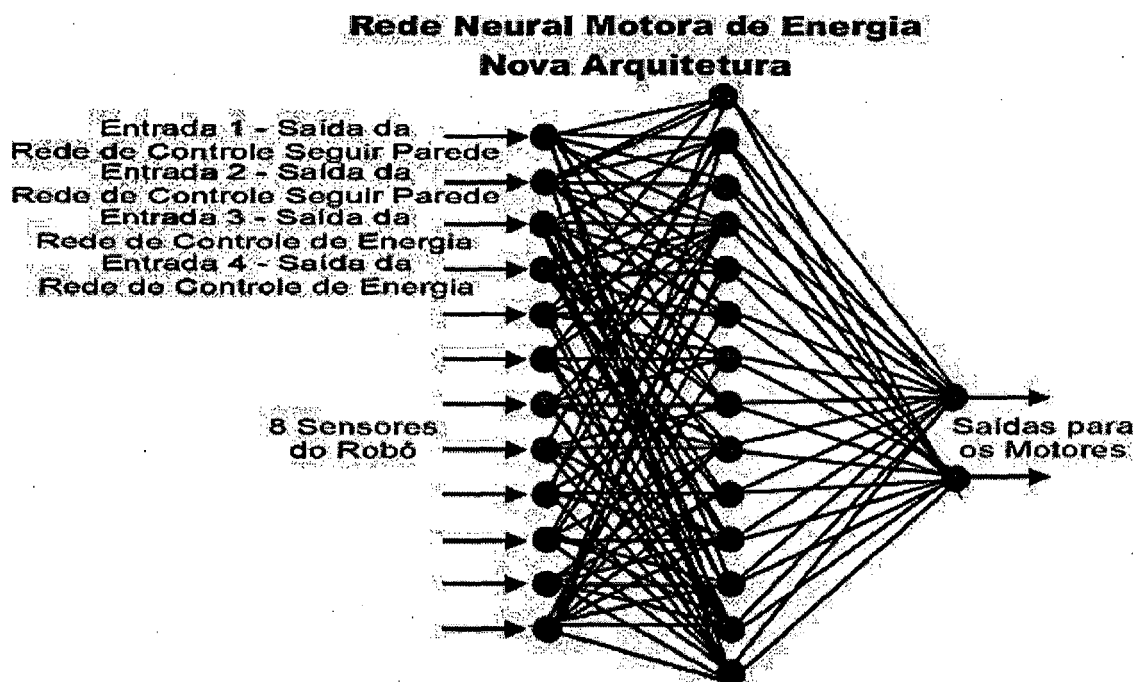


Figura 4.20: Rede Neural Direta Motora - Energia

4.2.3 Hierarquia dos Autômatos de Estado Finito

A Hierarquia também pode ser vista nos AEF, uma vez que, as Redes Neurais formam uma hierarquia e representam os AEF.



Figura 4.21: Separação em níveis da Nova Hierarquia

AEF-E é o Autômato de Estado Finito do Controle de Energia, o AEF-SP: é o Autômato de Estado Finito da Rede de Controle Seguir Parede e, por fim, o AEF-PE é o Autômato de Estado Finito da Rede de Controle Energia.

Abaixo está a Hierarquia dos Autômatos de Estado Finito:

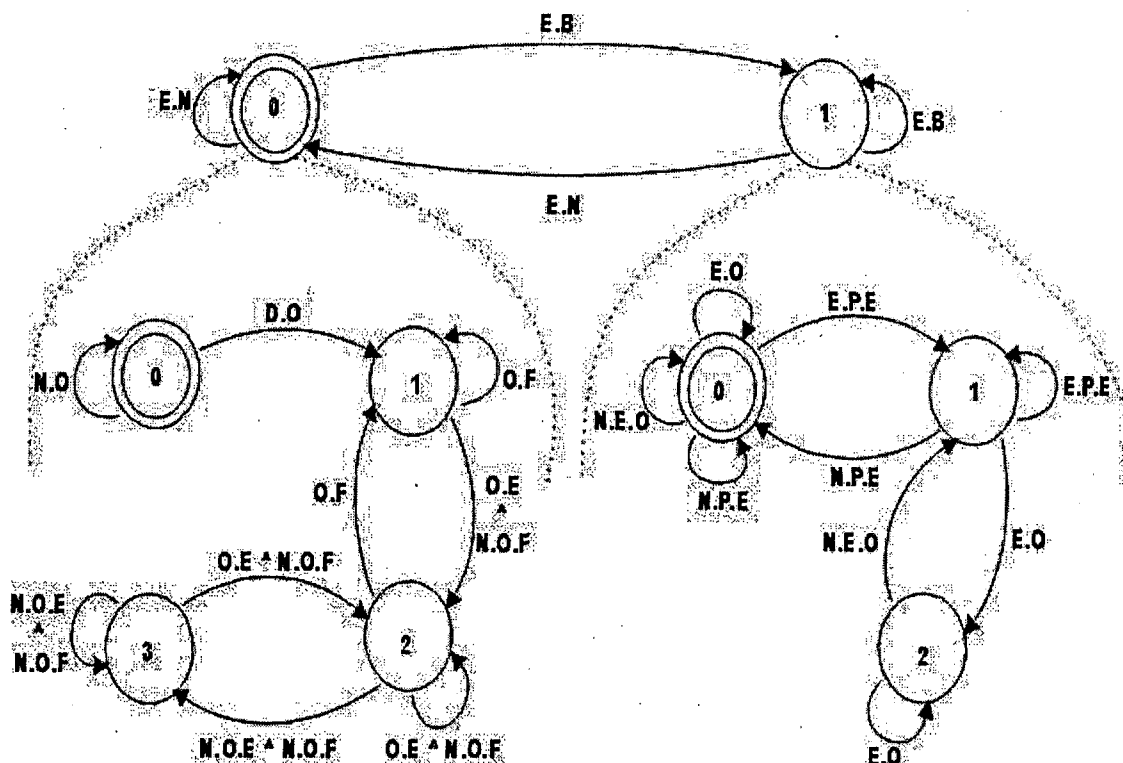


Figura 4.22: Hierarquia dos Autômatos de Estado Finito (AEF)

As figuras acima visam mostrar que as Redes de Controle estão ligadas ao Controle de Energia, isto é, a Rede de Controle Seguir Parede está ligada ao estado 0 (zero) e a Rede de Controle de Energia está ligada ao estado 1 (um) do Controle de Energia. Sendo assim, as duas redes de controle dependem do que ocorre no Controle de Energia para exercerem o controle sobre as redes motoras.

5 TREINAMENTO DAS REDES DA ARQUITETURA PARA CONTROLE DOS COMPORTAMENTOS REFLEXIVOS E REATIVOS

5.1.1 Definições e o Aprendizado da Rede Hierárquica para Controle dos Comportamentos Reflexivos e Reativos

“Trabalhos recentes na área de AA’s procuram construir dispositivos e dotá-los de um conjunto de comportamentos que os tornem aptos a realizarem as tarefas a que foram projetados. No entanto, dotar um AA de um conjunto fixo de comportamentos rígidos não é o suficiente para garantir a sua operação em um ambiente dinâmico como o mundo real. É necessário dotá-lo também da capacidade de selecionar, alterar e aprimorar seus comportamentos, tornando-o mais apto para operar no ambiente em que for colocado. A esta capacidade chamamos de ‘Aprendizado’.” (ROISENBERG, 1998).

O conjunto de treinamento é importante a ponto de merecer estudos e novas propostas, como podemos ver em NASCIMENTO & SILVA (1997). No simulador do robô de Khepera será implementada, primeiramente, uma rede neural para controlar os sensores de distância, chamada de Rede Sensória de Distância (KhepDist).

5.1.1.1 Rede Sensória de Distância

Para o treinamento “offline” e para obter os pesos criou-se uma rede fora do Khepera. Somente após o treinamento é que foram copiados os arquivos necessários para a execução do robô simulado. No treinamento utilizou-se o algoritmo de retropropagação. O intuito desta rede é de não permitir que o robô simulado colida com qualquer tipo de objeto, que para ele é considerado como obstáculo.

A arquitetura desta rede direta, como vimos, contém 8 neurônios na camada de entrada, representando os 8 sensores de distância, 8 neurônios na camada escondida e 2 neurônios na camada de saída, representando os motores.

Esta rede receberá os valores detectados pelos sensores de distância e, através de um cálculo, serão normalizados em valores que a rede “entenda”, isto é, em -1 para ausência e em 1 para presença de obstáculos. Por fim acionará os motores, podendo imprimir uma velocidade para avançar como também para recuar. O valor desta força é indicado por flechas vermelhas desenhadas no motor do robô simulado. Se a flecha estiver apontando para frente o robô estará avançando e no caso contrário, estará recuando. Para a normalização dos valores será usada a equação (4) mostrada abaixo:

$$((2.0 * \text{robot} \rightarrow \text{IRSensor}[i].\text{DistanceValue}) - (\text{MAXDIST} + \text{MINDIST})) / (\text{MAXDIST} - \text{MINDIST}) \quad (4)$$

Onde $\text{Robot} \rightarrow \text{IRSensor}[i].\text{DistanceValue}$ é a leitura dos sensores de distância e “i” é incrementado de 1 até 8, que são os sensores. MAXDIST é o valor limite dos sensores, significando colisão, e MINDIST é o menor valor dos sensores.

Por exemplo, se o valor detectado for 950 o cálculo resultará em 0,9, que está próximo de 1. Inicialmente foi feito um teste sem o controle da rede neural implementada para que pudéssemos determinar os valores dos sensores nos momentos críticos.

Estes momentos são considerados onde o robô encontra-se longe de qualquer obstáculo ou parede, a uma distância pequena, e quando ele tem uma colisão com qualquer coisa.

Neste teste o robô simulado “andou” pelo mundo e os sensores foram detectando o que estava a sua volta. Então, quando os sensores estiverem variando entre 0 e 500, significa que o robô simulado está longe de qualquer obstáculo ou parede, resultando, no avanço dos motores.

Se os valores estiverem entre 850 e 950 significa que o robô simulado está muito perto de um obstáculo ou parede, e neste caso, será verificado em qual, ou quais sensores estes valores aparecem. Para valores intermediários, entre 500 e 850, o robô adota o comportamento de manter os motores avançando. E para valores acima de 950 é considerado como colisão.

Com esta fase terminada, passamos para o treinamento da rede neural. Para este treinamento “offline”, foi preparado um conjunto de entradas com um número inicial de 34 situações em que o robô poderia se encontrar, isto é, fizemos a combinação de algumas situações possíveis dos sensores, somente para iniciar os testes. Estas entradas foram gravadas no arquivo chamado Khepdist.ent.

Tabela 5.1: Conjunto de treinamento da Rede Neural Direta – Sensores de Distância

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
1	-1	-1	-1	-1	-1	-1	-1
-1	1	-1	-1	-1	-1	-1	-1
-1	-1	1	-1	-1	-1	-1	-1
-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	1	-1	-1	-1
-1	-1	-1	-1	-1	1	-1	-1
-1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	1
-1	-1	-1	-1	-1	-1	-1	-1
1	1	-1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	-1	-1	-1
-1	-1	1	1	-1	-1	-1	-1
-1	-1	-1	1	1	-1	-1	-1
-1	-1	-1	-1	1	1	-1	-1
-1	-1	-1	-1	-1	1	1	-1
-1	-1	-1	-1	-1	-1	1	1
1	1	1	-1	-1	-1	-1	-1
-1	1	1	1	-1	-1	-1	-1
-1	-1	1	1	1	-1	-1	-1
-1	-1	-1	1	1	1	-1	-1
-1	-1	-1	-1	1	1	1	-1
-1	-1	-1	-1	-1	1	1	1
1	1	1	1	-1	-1	-1	-1
-1	1	1	1	1	-1	-1	-1
-1	-1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	1	-1
-1	-1	-1	-1	1	1	1	1
1	1	1	1	1	-1	-1	-1
-1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	-1
-1	-1	-1	1	1	1	1	1
1	1	-1	-1	1	1	-1	-1
-1	1	1	-1	1	1	-1	-1
-1	1	1	-1	-1	1	1	-1

Considerações a serem feitas, para os sensores:

- 1) -1 → Assume-se os valores entre 0 e 849.
- 2) 1 → Assume-se os valores entre 850 e 1023(limite).

O arquivo de exemplos, com as entradas e as saídas necessárias para o treinamento, está sendo apresentado na tabela a seguir, sendo S(i) os sensores (entradas) e Flag, a saída. Este arquivo foi gravado com o nome de Khepdist.dat

Tabela 5.2: Conjunto de exemplos – Entradas e saídas.

S.1	S.2	S.3	S.4	S.5	S.6	S.7	S.8	Flag
-1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-1	-1	1
-1	1	-1	-1	-1	-1	-1	-1	1
-1	-1	1	-1	-1	-1	-1	-1	1
-1	-1	-1	1	-1	-1	-1	-1	1
-1	-1	-1	-1	1	-1	-1	-1	1
-1	-1	-1	-1	-1	1	-1	-1	1
-1	-1	-1	-1	-1	-1	1	-1	1
-1	-1	-1	-1	-1	-1	-1	1	1
1	1	-1	-1	-1	-1	-1	-1	1
-1	1	1	-1	-1	-1	-1	-1	1
-1	-1	1	1	-1	-1	-1	-1	1
-1	-1	-1	1	1	-1	-1	-1	1
-1	-1	-1	-1	1	1	-1	-1	1
-1	-1	-1	-1	-1	1	1	-1	1
-1	-1	-1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	-1	-1	1
-1	1	1	1	-1	-1	-1	-1	1
-1	-1	1	1	1	-1	-1	-1	1
-1	-1	-1	1	1	1	-1	-1	1
-1	-1	-1	-1	1	1	1	-1	1
-1	-1	-1	-1	-1	1	1	1	1
1	1	1	1	-1	-1	-1	-1	1
-1	1	1	1	1	-1	-1	-1	1
-1	-1	1	1	1	1	-1	-1	1
-1	-1	-1	1	1	1	1	-1	1
-1	-1	-1	-1	1	1	1	1	1
1	1	1	1	1	-1	-1	-1	1
-1	1	1	1	1	1	-1	-1	1
-1	-1	1	1	1	1	1	-1	1
-1	-1	-1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1
-1	1	1	1	1	1	1	1	1
-1	-1	-1	1	1	1	1	1	1
1	1	1	-1	1	1	-1	1	1
-1	1	1	-1	1	1	-1	-1	1
-1	1	1	-1	-1	1	1	-1	1

Com estes dois arquivos já prontos passamos para o treinamento, no qual criamos o arquivo de definição da rede neural, chamado Khepdist.bpn. Neste arquivo definimos o número de neurônios em todas as camadas além de outros dados, como mostrado abaixo.

8 → Número de neurônios na camada de entrada

8 → Número de neurônios na camada oculta

1 → Número de neurônios na camada de saída

0.9 → Valor de ALPHA - momentum

0.1 → Valor de ETA - learning rate

34 → Número de exemplos no arquivo de exemplos

Este arquivo é lido em primeiro lugar. Após ele, serão lidos os arquivos de exemplos e de entradas, dando início ao treinamento e, assim que terminar, é necessário gravar os pesos calculados, no arquivo de pesos. Este arquivo terá o nome de Khepdist.wgt.

Para podermos entender o que ocorrerá quando o robô simulado estiver em funcionamento, devemos ver, na teoria, as seguintes considerações:

- 1) Quando qualquer sensor detectar um obstáculo o seu valor será 1, com isto, o valor do Flag será 1.
- 2) Se nenhum dos sensores detectar obstáculo o valor será -1 e o do Flag também será -1.

Nossa proposta diz respeito ao controle de comportamentos, portanto, apenas 1(um) não é suficiente para sua validação, o que levou à implementação do outro comportamento que é o “Encontrar Foco Luminoso”.

5.1.1.2 Rede Sensória de Luz

Esta rede (comportamento) controla os sensores de luminosidade, a qual é chamada de Rede Sensória de luz (KhepLuz). Ela também está presente no robô real e no simulado. Sua configuração é idêntica a dos sensores de distância, sendo 8 sensores distribuídos e posicionados juntos aos de distância. Seu treinamento será igual ao da rede neural de controle dos sensores de distância, será em “offline” e com o algoritmo de retropropagação. A diferença está nos arquivos de treinamentos assim como nos valores de detecção, os quais serão mostrados a seguir.

A variação dos valores dos sensores de luminosidade, para uma melhor compreensão, dá-se de forma contrária ao dos sensores de distância, isto é, enquanto os

sensores de distância variam de 0 (ausência de obstáculo) para aproximadamente 1023 (colisão com obstáculo), o de luminosidade varia de 0 (presença forte de luz) a aproximadamente 600 (ausência total de luz). Da mesma forma que a outra rede, esta terá que receber os valores dos sensores e normalizá-los em -1 (presença de luz) e 1 (ausência de luz) e, para isto, será usada a equação (5), abaixo mostrada:

$$((2.0 * \text{robot} \rightarrow \text{IRSensor}[i].\text{LightValue}) - (\text{MAXLUZ} + \text{MINLUZ})) / (\text{MAXLUZ} - \text{MINLUZ}) \quad (5)$$

O cálculo, da mesma forma que o cálculo dos valores da distância, normalizará os valores detectados pelos sensores para os valores que a rede trabalhará, diferenciando apenas a variável, de "DistanceValue" para "LightValue".

Por exemplo, se um dos sensores estiver com o valor 545 (quinhentos e quarenta e cinco), a conversão trará para a rede o valor 0,9818182, significando ausência de luz, pois está próximo de 1. O arquivo de entradas, Khepluz.ent, inicialmente tem a seguinte configuração:

Tabela 5.3: Conjunto de treinamento para a Rede Neural Direta – Sensores de Luminosidade

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8
1	1	1	1	1	1	1	1
-1	-1	1	1	1	1	1	1
1	-1	-1	1	1	1	1	1
1	1	-1	-1	1	1	1	1
1	1	1	-1	-1	1	1	1
1	1	1	1	-1	-1	1	1
1	1	1	1	1	-1	-1	1
1	1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	1	1
1	-1	-1	-1	1	1	1	1
1	1	-1	-1	-1	1	1	1
1	1	1	-1	-1	-1	1	1
1	1	1	1	-1	-1	-1	1
1	1	1	1	1	-1	-1	-1

O conjunto de entradas para o treinamento é pequeno, pois se trata de um foco luminoso e devido à configuração do robô é impossível ativar apenas 1(um) sensor de cada vez, uma vez que a luz dispersada atinge, sempre, 2 ou mais sensores.

Como para o teste será colocado, no mundo, apenas um foco luminoso, será

impossível ativar todos os sensores de uma vez. Apenas os dois sensores frontais irão ter seus valores totais atingidos, pois, a tendência do robô é sempre estar de frente com o foco luminoso. O conjunto de exemplos para o treinamento da rede neural de luminosidade foi definido e é apresentado na tabela a seguir, sendo $S(i)$ os sensores e Flag, a saída.

Tabela 5.4: Conjunto de exemplos – Entradas e Saídas

S.1	S.2	S.3	S.4	S.5	S.6	S.7	S.8	Flag
-1	-1	1	1	1	1	1	1	1
1	-1	-1	1	1	1	1	1	1
1	1	-1	-1	1	1	1	1	1
1	1	1	-1	-1	1	1	1	1
1	1	1	1	-1	-1	1	1	1
1	1	1	1	1	-1	-1	1	1
1	1	1	1	1	1	-1	-1	1
-1	-1	-1	1	1	1	1	1	1
1	-1	-1	-1	1	1	1	1	1
1	1	-1	-1	-1	1	1	1	1
1	1	1	-1	-1	-1	1	1	1
1	1	1	1	-1	-1	-1	1	1
1	1	1	1	1	-1	-1	-1	1
1	1	1	1	1	1	1	1	-1

Podemos considerar:

- 1) Se qualquer dos sensores estiver sentindo a presença de luz o valor deste será -1 e o Flag será 1.
- 2) Se nenhum dos sensores detectar a presença de luz, o seu valor será 1 e o Flag será -1.

Todo o processo está sendo idêntico ao da rede de distância. Isto significa que já se pode fazer o treinamento desta rede, sendo necessário criar o arquivo que define a rede neural, chamado Khepluz.bpn. Neste arquivo definimos os dados, número de neurônios nas camadas, como se segue:

8 → Número de neurônios na camada de entrada

8 → Número de neurônios na camada escondida

1 → Número de neurônios na camada de saída

0.9 → Valor de ALPHA - momentum

0.1 → Valor de ETA - learning rate

14 → Número de exemplos no arquivo de exemplos

Estas duas redes fazem parte do primeiro nível da hierarquia e faltam definir as outras duas que também fazem parte deste nível, mas com função de controlar os motores.

5.1.1.3 Rede Motora de Distância – Comportamento Reflexivo

As duas redes responsáveis pelo controle dos motores também terão como entrada os sensores do robô, cada uma com sua função (desviar de obstáculo e ir ao encontro do foco). Uma das redes tem o nome de Rede Motora de Distância (KhepMotD), que tem em seu significado o controle do motor com a influência dos sensores de distância. E a outra rede tem o nome de Rede Motora de Luz (KhepMotL), que terá seu controle do motor com a influência dos sensores de luminosidade.

Elas obedecerão ao mesmo esquema de treinamento das duas redes anteriores, mas tendo sua diferenciação no conjunto de treinamento, tanto no arquivo de entrada como no arquivo de exemplos.

Dispensaremos aqui a apresentação de todo o conjunto de treinamento, uma vez que nos exemplos acima podemos ver como este processo foi feito, todavia será mostrada uma parte dos arquivos para entendermos o funcionamento das redes.

Para isto usaremos primeiro o exemplo da rede KhepMotD. Segue abaixo a configuração da rede através do arquivo KhepMotD.bpn.

10 → Número de neurônios na camada de entrada

10 → Número de neurônios na camada escondida

2 → Número de neurônios na camada de saída

0.9 → Valor de ALPHA - momentum

0.1 → Valor de ETA - learning rate

102 → Número de exemplos no arquivo de exemplos

A definição mostra uma diferença em relação às outras duas redes, aumentaram os neurônios, principalmente, na camada de entrada e na de saída. Este aumento na camada de entrada é devido à saída da rede neural de controle, pois suas saídas são entradas para esta rede, e o aumento na saída é referente aos dois motores do robô.

A rede de controle está posicionada em um nível acima e será a última a ser definida. Como a rede de controle representa um autômato de estado finito (AEF), suas saídas representarão os estados deste autômato. Em DE AZEVEDO (1997) e MERLAT et al (1997) podemos ver que existe uma relação entre um “autômato” e as RNA’s.

Nesta rede KhepMotD, não bastará ser detectado um obstáculo, mas terá que ser analisado o estado em que se encontra a rede de controle. Segue, então, uma parte do conjunto de entradas da rede KhepMotD para entendermos como ela funcionará.

Tabela 5.5: Conjunto de treinamento para a Rede Neural Direta – KhepMotD

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sc1	Sc1
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
-1	-1	-1	-1	-1	-1	-1	-1	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	-1	-1	-1	1	-1
1	-1	-1	-1	-1	-1	-1	-1	-1	1
1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	1	-1	-1	-1	-1	-1	-1	1	-1
1	1	-1	-1	-1	-1	-1	-1	-1	1
1	1	1	-1	-1	-1	-1	-1	-1	-1
1	1	1	-1	-1	-1	-1	-1	1	-1
1	1	1	-1	-1	-1	-1	-1	-1	1
1	1	1	1	-1	-1	-1	-1	-1	-1
1	1	1	1	-1	-1	-1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	1
1	1	1	1	1	-1	-1	-1	-1	-1
1	1	1	1	1	-1	-1	-1	1	-1
1	1	1	1	1	-1	-1	-1	-1	1

A tabela está reduzida, mostrando apenas os primeiros sensores com valores 1, ela estando completa, terá um total 102 combinações de entradas. Lembrando que o conjunto de entradas é formado através da combinação das situações possíveis entre os sensores e os estados da rede de controle.

Importante perceber nesta tabela as duas últimas entradas referentes às saídas da

Rede de Controle (Sc1 e Sc2), onde Sc1 nomeada como Saída 1 da Rede de controle e Sc2 Saída dois.

Neste nosso caso, as saídas representam uma AEF com 3 estados, como visto na definição da rede recorrente. Aqui o importante é ver que, para cada estado, temos a mesma configuração de entradas. Exemplo:

Tabela 5.6: Exemplo do conjunto de entradas para a rede KhepMotD.

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sc1	Sc1
-1	1	1	-1	-1	-1	-1	-1	-1	-1
-1	1	1	-1	-1	-1	-1	-1	1	-1
-1	1	1	-1	-1	-1	-1	-1	-1	1

Por este exemplo podemos ver que as entradas Sc1 e Sc2 variam, enquanto as entradas dos sensores permanecem as mesmas. O conjunto de exemplos terá a mesma quantidade do arquivo de entradas, juntando-se a ele as saídas desta rede que são referentes aos motores do robô simulado.

Tabela 5.7: Conjunto de Exemplos – Rede KhepMotD.

S1	S2	S3	S4	S5	S6	S7	S8	Sc1	Sc1	ME	MD
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	1
-1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	1
-1	-1	-1	-1	-1	-1	-1	-1	-1	1	0	0
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
1	-1	-1	-1	-1	-1	-1	-1	1	-1	1	-1
1	-1	-1	-1	-1	-1	-1	-1	-1	1	0	0
1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
1	1	-1	-1	-1	-1	-1	-1	1	-1	1	-1
1	1	-1	-1	-1	-1	-1	-1	-1	1	0	0
1	1	1	-1	-1	-1	-1	-1	-1	-1	1	-1
1	1	1	-1	-1	-1	-1	-1	1	-1	1	-1
1	1	1	-1	-1	-1	-1	-1	-1	1	0	0
1	1	1	1	-1	-1	-1	-1	-1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	1	0	0
1	1	1	1	1	-1	-1	-1	-1	-1	1	-1
1	1	1	1	1	-1	-1	-1	1	-1	1	-1
1	1	1	1	1	-1	-1	-1	-1	1	0	0

Considerações:

- 1) S(i): São os sensores do robô.
- 2) Sc(x): São as saídas da rede de controle e entradas para a rede KhepMotD.
- 3) ME: Saída da rede KhepMotD que atua sobre o Motor Esquerdo do Robô.
- 4) MD: Saída da rede KhepMotD que atua sobre o Motor Direito do Robô.

Com esta rede definida, fica faltando apenas a rede KhepMotL para completar o nível inferior da hierarquia. Esta rede ficará responsável pela implementação do controle dos motores que são influenciados pelos sensores de luminosidade.

5.1.1.4 Rede Motora de Luminosidade – Comportamento Reflexivo

Da mesma forma que a rede KhepMotD, a configuração desta terá o mesmo número de neurônios nas camadas, com diferenciação apenas no número de exemplos. Seu arquivo terá o nome de KhepMotL.bpn. Sua configuração é mostrada a seguir:

- 10 → Número de neurônios na camada de entrada
- 10 → Número de neurônios na camada escondida
- 2 → Número de neurônios na camada de saída
- 0.9 → Valor de ALPHA - momentum
- 0.1 → Valor de ETA - learning rate
- 81 → Número de exemplos no arquivo de exemplos

Com relação aos conjuntos de treinamentos, será adotado o mesmo procedimento usado para a rede KhepMotD. Assim, podemos ver a seguir o conjunto de entrada e de exemplos, para ser feita a comparação com os outros arquivos.

Tabela 5.8: Conjunto de Entradas – Rede KhepMotL.

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sc1	Sc1
1	1	1	1	1	1	1	1	-1	-1
1	1	1	1	1	1	1	1	1	-1
1	1	1	1	1	1	1	1	-1	1
-1	1	1	1	1	1	1	1	-1	-1
-1	1	1	1	1	1	1	1	1	-1
-1	1	1	1	1	1	1	1	-1	1
-1	-1	1	1	1	1	1	1	-1	-1
-1	-1	1	1	1	1	1	1	1	-1
-1	-1	1	1	1	1	1	1	-1	1
-1	-1	-1	1	1	1	1	1	-1	-1
-1	-1	-1	1	1	1	1	1	1	-1
-1	-1	-1	1	1	1	1	1	-1	1
-1	-1	-1	-1	1	1	1	1	1	-1
-1	-1	-1	-1	1	1	1	1	-1	1
-1	-1	-1	-1	-1	1	1	1	-1	-1
-1	-1	-1	-1	-1	1	1	1	1	-1
-1	-1	-1	-1	-1	1	1	1	-1	1

Esta tabela também está reduzida, uma vez que ela tem 81(oitenta e uma) combinações de entradas. Nas combinações mostradas acima, pode-se perceber que, para cada conjunto das entradas Sc1 e Sc2, existe a mesma entrada dos sensores; Configurando-se 3 combinações para cada situação dos sensores de entradas, Como mostrado a seguir:

Tabela 5.9: Exemplo do conjunto de entradas – Rede KhepMotL.

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8	Sc1	Sc2
1	1	1	-1	-1	1	1	1	-1	-1
1	1	1	-1	-1	1	1	1	1	-1
1	1	1	-1	-1	1	1	1	-1	1

No arquivo de exemplos teremos todos os sensores e as entradas Sc1 e Sc2, assim como, as saídas desta rede que serão responsáveis pelo avanço ou recuo dos motores do robô.

Tabela 5.10: Conjunto de exemplos – Rede KhepMotL.

S1	S2	S3	S4	S5	S6	S7	S8	Sc1	Sc2	ME	MD
1	1	1	1	1	1	1	1	-1	-1	0	0
1	1	1	1	1	1	1	1	1	-1	0	0
1	1	1	1	1	1	1	1	-1	1	1	1
-1	1	1	1	1	1	1	1	-1	-1	0	0
-1	1	1	1	1	1	1	1	1	-1	0	0
-1	1	1	1	1	1	1	1	-1	1	-1	1
-1	-1	1	1	1	1	1	1	-1	-1	0	0
-1	-1	1	1	1	1	1	1	1	-1	0	0
-1	-1	1	1	1	1	1	1	-1	1	-1	1
-1	-1	-1	1	1	1	1	1	-1	-1	0	0
-1	-1	-1	1	1	1	1	1	1	-1	0	0
-1	-1	-1	1	1	1	1	1	-1	1	-1	1
-1	-1	-1	-1	1	1	1	1	-1	-1	0	0
-1	-1	-1	-1	1	1	1	1	1	-1	0	0
-1	-1	-1	-1	1	1	1	1	-1	1	-1	1
-1	-1	-1	-1	-1	1	1	1	-1	-1	0	0
-1	-1	-1	-1	-1	1	1	1	1	-1	0	0
-1	-1	-1	-1	-1	1	1	1	-1	1	-1	1

Considerações:

- 1) S(i): São os sensores do robô.

- 2) Sc(x): São as saídas da rede de controle e entradas para a rede KhepMotL.
- 3) ME: Saída da rede KhepMotL e que atua sobre o Motor Esquerdo do Robô.
- 4) MD: Saída da rede KhepMotL e que atua sobre o Motor Direito do Robô.

5.1.1.5 Rede de Controle – Comportamentos Reativos

Por fim, iniciamos a definição da Rede Neural de Controle, chamada de Rede de Controle (KhepRec). Ela é a principal rede da hierarquia, pois terá que controlar as redes KhepMotD e KhepMotL. Por ela controlar as outras, deverá ter memória, e isto será possibilitado pela sua topologia que é recorrente.

Esta rede receberá as saídas das redes KhepDist e KhepLuz, indicando se foi encontrado obstáculo ou luz. Receberá, também, as suas próprias saídas (realimentação), com isto ela poderá controlar as redes motoras.

Tem-se na configuração desta rede:

- 4 → Número de neurônios na camada de entrada
- 4 → Número de neurônios na camada escondida
- 2 → Número de neurônios na camada de saída
- 0.9 → Valor de ALPHA - momentum
- 0.1 → Valor de ETA - learning rate
- 12 → Número de exemplos no arquivo de exemplos

Os conjuntos de treinamento, KhepRec.ent e KhepRec.dat, serão listados a seguir, para vermos como ela fará o controle das outras redes. E diferentemente dos outros conjuntos de treinamento, este está completo.

Todas as situações possíveis assumidas pela rede de controle, ou , todos os estados que a rede irá assumir se baseiam nas saídas das redes sensórias e nas suas próprias saídas (realimentação). E os estados são: Vagar/Desviar, Encontrou Luz e Encontrou Obstáculo.

Tabela 5.11: Conjunto de entradas – KhepRec.ent

Ent1	Ent2	Ent3	Ent4
-1	-1	-1	-1
1	-1	-1	-1
-1	1	-1	-1
1	1	-1	-1
-1	-1	1	-1
1	-1	1	-1
-1	1	1	-1
1	1	1	-1
-1	-1	-1	1
1	-1	-1	1
-1	1	-1	1
1	1	-1	1

Ent1 é a saída da rede KhepDist e Ent2 a saída da rede KhepLuz. As outras duas entradas (Ent3 e Ent4) são referentes à realimentação.

Tabela 5.12: Conjunto de entradas – KhepRec.dat

Ent1	Ent2	Ent3	Ent4	Saída1	Saída2
-1	-1	-1	-1	-1	-1
1	-1	-1	-1	-1	1
-1	1	-1	-1	-1	-1
1	1	-1	-1	-1	-1
-1	-1	1	-1	-1	1
1	-1	1	-1	-1	1
-1	1	1	-1	1	-1
1	1	1	-1	1	-1
-1	-1	-1	1	-1	-1
1	-1	-1	1	-1	1
-1	1	-1	1	1	-1
1	1	-1	1	1	-1

Com esta definição a hierarquia está completa. Como nos casos anteriores, após o treinamento será gravado o arquivo, contendo os pesos, com o nome de KhepRec.wgt. Para os testes no Simulador do Robô de Khepera basta copiarmos os arquivos de definição da rede “.bpn” e os arquivos de pesos “.wgt” e executarmos o ambiente de simulação.

5.1.2 Análise dos Resultados

Nesta seção procuraremos descrever o que realmente aconteceu com o robô simulado, após as implementações acima citadas, terem sido colocadas no simulador. As etapas desta parte foram:

- I. O robô foi colocado para circular em um mundo que não havia nada, além das quatro paredes.
- II. Na segunda fase ele foi colocado para circular em um mundo que tinha alguns obstáculos e paredes formando um tipo de beco.
- III. Nesta terceira fase, o mundo foi composto com as quatro paredes e um foco luminoso.
- IV. Na quarta, o robô circulou em um mundo composto por paredes, obstáculos e um foco luminoso longe de qualquer obstáculo ou parede.
- V. Por fim, um mundo composto por obstáculos, paredes e um foco luminoso encostado atrás de um obstáculo.

Estas 5 situações serão um teste para o robô, se ele transpuser todas sem problemas, iremos aumentar a complexidade do mundo cada vez mais, na tentativa de não termos problemas em qualquer que seja o mundo.

Para um teste preliminar, a rede de distância foi treinada e posta para executar sua tarefa, mas encontrou situações onde o comportamento resultante apresentou o robô rodando em círculos.

Da mesma forma, implementamos a rede para reconhecer e ir de encontro à luz e desabilitamos a rede de distância. Ativamos os sensores de luminosidade e o robô, quando encontrava um ponto de luz atrás de um obstáculo, chocava-se com ele para poder cumprir a sua tarefa. Este processo foi feito para que pudéssemos fazer comparações entre os comportamentos resultantes antes e depois da implementação da hierarquia.

Sabemos que um problema emerge quando vários comportamentos, muitos inclusive antagônicos, são implementados e postos para operar simultaneamente, uma vez que, não se sabe qual comportamento emergirá quando, por exemplo, o robô estiver diante de um obstáculo e ao mesmo tempo procurar seguir uma fonte de luz.

Uma solução para estes problemas é a utilização desta hierarquia, na qual se conecta as redes neurais diretas, responsáveis pela implementação dos comportamentos reflexivos, a uma rede recorrente, para que esta faça o controle de qual comportamento é mais adequado a cada momento (MCFARLAND & BÖSSER, 1993).

Com relação aos testes feitos com esta hierarquia implementada, na primeira fase, descrita na parte (I), o robô desviou das paredes e circulou pelo mundo. Esta fase não apresentou um grau médio de complexidade para a avaliação do desempenho do robô simulado.

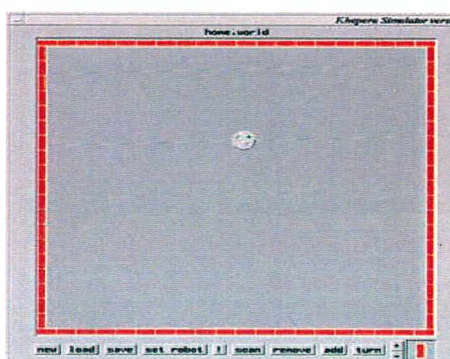


Figura 5.1: Fase I, teste sem obstáculos ou luz

Então passamos para a fase (II), com um pouco mais de complexidade. Nesta o robô não ficou preso no “beco”, como acontecia antes de implementarmos a hierarquia. Ele também não colidiu com nenhum obstáculo, parede ou muros colocados no meio do mundo, como apresenta a figura abaixo:

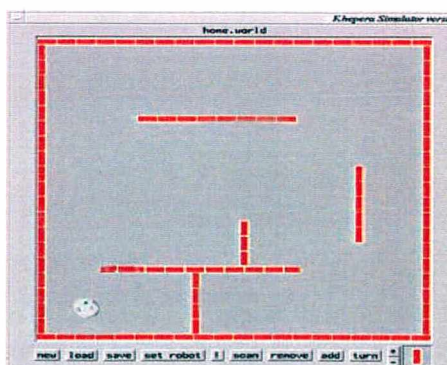


Figura 5.2: Fase II, teste com obstáculos

Fase (III). Conforme as fases vão sendo completadas, o mundo vai sendo

modificado, com isto, nesta fase mudaremos o enfoque, que eram os sensores de distância, para os sensores de luminosidade.

Nesta fase o robô correu pelo mundo até encontrar o foco de luz, indo em sua direção. Mas o foco de luz é um “abajur” e, por isto, no momento em que se aproximou muito, os sensores de distância foram ativados, desabilitando os de luminosidade.

Isto provocou um distanciamento do robô em relação ao foco de luz, contudo, quando ele estava a uma distância na qual seus sensores de distância não detectaram o foco como obstáculo e sim como luz, ele habilitou a rede de luminosidade e desabilitou a de distância, ocasionando seu retorno ao foco de luz. E assim ele ficou, rodeando o foco a cada mudança de estado.

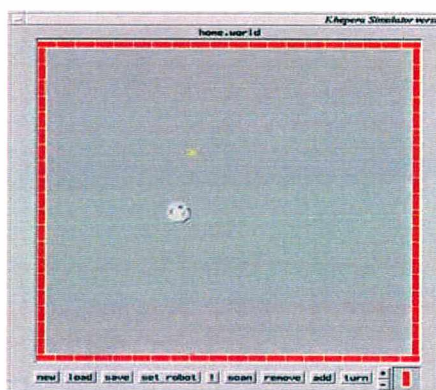


Figura 5.3: Fase III, teste para encontrar luz.

Na penúltima fase dos testes, a IV (quarta), teremos na configuração do mundo, obstáculos e foco de luz, para que a complexidade seja maior. Porém, o foco de luz foi posicionado longe de qualquer obstáculo, pois o objetivo foi ver se o robô conseguiria desviar de obstáculos e se quando encontrasse o foco luminoso seu estado passaria a ser o estado 1 (encontrou luz), indo direto para ele.

Iniciamos os testes colocando o robô em uma posição que não tivesse nenhum obstáculo ou foco de luz sendo sentido por seus sensores. Quando o colocamos para rodar, ele assumiu uma rota reta até o primeiro obstáculo encontrado, e desviou-se do mesmo, tomando uma outra rota também de colisão, mas em um dos seus sensores fi

detectada a presença do foco de luz, ele virou e seguiu o foco. Quando chegou perto demais deste foco, assumiu o mesmo comportamento apresentado na fase III.

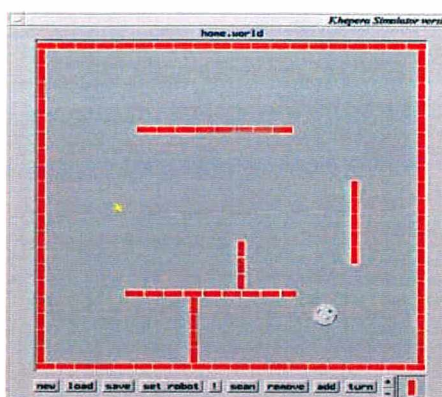


Figura 5.4: Fase IV, teste com obstáculo e foco de luz.

Na última fase, fase V, o mundo foi construído para testar a capacidade da hierarquia de controlar o robô, pois o foco de luz foi colocado encostado em um “muro”, não possibilitando ao robô ficar rodeando o ponto de luz.

O mais importante acontecimento foi o momento em que os sensores do robô detectaram o foco luminoso através da parede translúcida e ele foi em sua direção, mas quando os sensores de distância detectaram a parede, o estado que era 1 (encontrou Luz) passou para o estado 2 (desviar), e o robô iniciou o desvio da parede. Figura abaixo.

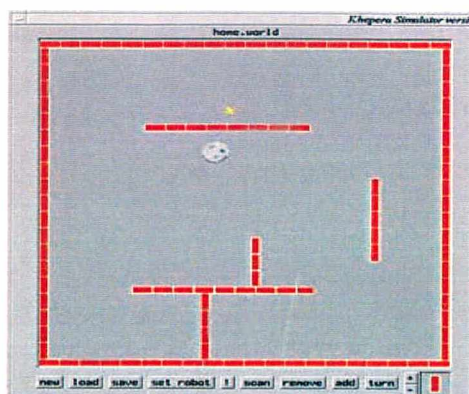


Figura 5.5: Fase V, teste com foco de luz atrás de um obstáculo

Como a parede era grande em relação ao foco, conforme o robô foi seguindo a parede, deixou de sentir o foco, tomando outra rota. Quando encontrava a luz, ele

encostava, mas sentia o obstáculo e saía.

5.1.3 Definições e Aprendizado da Nova Arquitetura para Controle dos Comportamentos Reflexivos, Reativos e Controle de Energia

Como visto na seção anterior, os comportamentos implementados para o início dos testes da RNAH foram satisfatórios, porém não atingindo um grau de inteligência pretendido por este trabalho. Isto nos levou a uma nova implementação que irá incluir mais duas redes, totalizando 7 (sete) redes neurais.

As redes estão divididas da seguinte forma: 3 (três) redes diretas no Subsistema Sensorio, 2 (duas) redes recorrentes no subsistema de Controle, 2 (duas) redes diretas no Subsistema Motor. Também está incluso um Perceptron que é o responsável pelo controle do “nível de energia”. As definições do conjunto de treinamento de cada rede e do perceptron estão a seguir.

No caso do Controle de Energia não é necessário colocar o conjunto de treinamento. Na seção 4.2.2.3 apresenta-se a equação e o algoritmo pertinente ao perceptron, no que tange sua função em controlar a “energia” do robô simulado.

5.1.3.1 Conjunto de Treinamento para a Rede Sensória Seguir Parede

O Conjunto de treinamento representa as situações possíveis das saídas, isto significa que, em alguns casos, somente uma das saídas terá seu valor em 1 (Ativo) e as demais em -1 (inativo). Em outros casos, duas ou mais terão seus valores em 1 significando que seus sensores detectaram mais de um obstáculo ou o encontro de duas paredes, um canto.

A tabela a seguir mostra uma parte dessas situações, já que o conjunto todo é grande para ser mostrado aqui. Assim como ocorreu na apresentação dos conjuntos de treinamento da arquitetura anterior, neste também será mostraremos parte do conjunto de treinamento, porém ele dará uma noção de como podemos montá-lo por inteiro, isto é, pegando todas as entradas e saídas e fazendo sua combinação.

Tabela 5.13: Conjunto de Treinamento da Rede Sensória Seguir Parede.

Entradas								Saídas				
S1	S2	S3	S4	S5	S6	S7	S8	OF	OE	NOF	NOE	NO
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	1
1	1	-1	-1	-1	-1	-1	-1	-1	1	-1	-1	-1
-1	-1	1	1	-1	-1	-1	-1	1	-1	-1	-1	-1
-1	-1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1
-1	1	1	-1	-1	-1	-1	-1	1	1	-1	-1	-1

Sx representam os 8 (oito) sensores de distância do robô e os valores -1 e 1 representam, respectivamente, ausência e presença de obstáculo. O par S1 e S2 está na parte Esquerda do Robô, S3 e S4 na parte Frontal, S5 e S6 na parte direita, S7 e S8 na parte Traseira. Uma situação muito comum é a de dois sensores detectarem o obstáculo, por exemplo, sensores localizados na parte esquerda e na parte frontal (S2 e S3, respectivamente). A seguinte situação provoca duas saídas ativas, isto é, duas saídas assumirão o valor 1 (última linha da tabela 5.13) que representa atividade, e as demais assumirão o valor -1 (inatividade).

A situação apresentada acima levará o robô a virar para a direita, comportamento que visa contornar o obstáculo. Como as saídas dependem dos sensores de distância do robô, haverá casos em que alguns sensores detectarão o obstáculo e, por exemplo, se forem os sensores S2, S3, S4 e S5, o robô terá como informação a presença de obstáculo à esquerda (OE), frontal (OF) e à direita (NOE). Os nomes das saídas foram convencionados desta forma apenas para melhorar a definição do conjunto de treinamento. Ao invés de chamarmos de OD a presença de obstáculo à direita, optou-se por chamar de NOE (ausência de obstáculo à esquerda), já que o robô sempre irá procurar se posicionar com a “parede” a sua esquerda, para segui-la.

Por isso, ficou conveniente denominar de NOE as situações nas quais os sensores direitos e/ou traseiros indicarem a presença de obstáculo. Ocorrerá situação em que apenas 1 sensor indicará a presença de obstáculo e esse poderá ser um sensor traseiro, indicando que o obstáculo não é frontal e nem esquerdo.

Da mesma forma, poderá ocorrer situação na qual o obstáculo é sentido por um ou dois sensores da direita, significando que o obstáculo não está à esquerda e nem é

frontal (NOE/NOF).

Essa explicação ficará mais clara após a descrição da Rede de Controle Seguir Parede. Vemos, então, que a função desta rede é a de informar ao controle (Rede de Controle Seguir Parede) a posição da “parede” e/ou “obstáculo” para ele atuar sobre a rede motora.

5.1.3.2 Conjunto de Treinamento das Redes Sensórias de Distância e de Energia

Com visto na seção 4.2.2.2, os conjuntos de treinamento das redes sensórias de distância e de energia são os mesmos da arquitetura anterior. A rede sensória de distância receberá os valores detectados pelos sensores de distância e, através de um cálculo, serão normalizados nos valores -1 para ausência e 1 para presença de obstáculos.

Os valores dos sensores de energia variam de forma contrária aos dos valores dos sensores da rede de distância. Enquanto os sensores de distância variam de 0 (ausência de obstáculo) para aproximadamente 1023 , o de energia varia de 0 (presença forte de luz) a aproximadamente 600 (ausência total de luz). Da mesma forma que a outra rede, esta terá que receber os valores dos sensores e normalizá-los em -1 (presença de luz) e 1 (ausência de luz)

Tabela 5.14: Conjunto de treinamento da Rede Sensória de Distância

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Sensor 5	Sensor 6	Sensor 7	Sensor 8
1	-1	-1	-1	-1	-1	-1	-1
1	1	-1	-1	-1	-1	-1	-1
1	1	1	-1	-1	-1	-1	-1
1	1	1	1	-1	-1	-1	-1
1	1	1	1	1	-1	-1	-1

Este conjunto tem apenas os primeiros valores em 1 , para mostrar que o conjunto é o mesmo da arquitetura anterior.

Tabela 5.15: Conjunto de treinamento da Rede Neural Sensória de Energia

Sensor1	Sensor2	Sensor3	Sensor4	Sensor5	Sensor6	Sensor7	Sensor8
1	1	1	1	1	1	1	1
-1	-1	1	1	1	1	1	1
-1	-1	-1	1	1	1	1	1

Da mesma forma, este conjunto de treinamento está mostrando apenas algumas situações que já foram apresentadas no conjunto de treinamento da rede sensória de luminosidade, pois as suas são idênticas.

5.1.3.3 Conjunto de Treinamento da Rede de Controle Seguir Parede

A rede de controle recebe dados da rede sensória seguir parede e também do perceptron, para que ela possa controlar a rede motora de distância. Parte do conjunto de treinamento da Rede de Controle Seguir Parede está sendo mostrado a seguir.

Tabela 5.16: Conjunto de Treinamento da Rede de Controle Seguir Parede

Energia	OE	OF	NOE	NOF	NO	RRCSP1	RRCSP2	SRCSP1	SRCSP2
1	-1	-1	-1	-1	1	-1	-1	-1	-1
1	1	-1	-1	1	-1	-1	-1	1	-1
1	-1	1	-1	-1	-1	-1	-1	-1	1
1	-1	-1	1	1	-1	-1	-1	1	1
0	-1	-1	-1	-1	1	-1	-1	0	0
0	1	-1	-1	1	-1	-1	-1	0	0
0	-1	1	-1	-1	-1	-1	-1	0	0
0	-1	-1	1	1	-1	-1	-1	0	0

Energia é o valor extraído através do comando de seleção (IF) do Controle de Energia. RRCSP1 e RRCSP2 são as entradas referentes à retroação, uma vez que essa rede é do tipo recorrente e SRCSP1 e SRCSP2 são as saídas dessa mesma rede. As entradas OE, OF, NOE, NOF e NO são as saídas da rede sensória seguir parede, as quais informam a posição da “parede” ou do “obstáculo”.

Pelo conjunto de treinamento percebe-se que a atuação da Rede de Controle Seguir Parede depende da energia. Se a energia estiver baixa, quem irá atuar nas redes motoras é a Rede de Controle Buscar Energia.

Os estados do Autômato são representados pelos conjuntos: $\{-1,-1\}$ estado 0 (zero), $\{1,1\}$ estado 1 (um), $\{1,-1\}$ estado 2 (dois) e $\{-1,1\}$ estado 3 (três). Sua função é de manter o robô seguindo, pela esquerda, uma parede enquanto sua energia estiver “normal”.

5.1.3.4 Conjunto de Treinamento da Rede de Controle de Energia

Esta rede exercerá controle sobre as duas redes motoras, já que ela terá que manter o robô desviando de obstáculos enquanto procura o foco de energia para poder se reabastecer. Ela receberá informação do perceptron e das duas redes sensórias (Rede Sensória de Distância – RSD e Rede Sensória de Energia - RSE). Parte do conjunto de treinamento segue abaixo:

Tabela 5.17: Conjunto de Treinamento da Rede de Controle de Energia

Energia	SRSE	SRSD	RRCE1	RRCE2	SRCE1	SRCE2
0	-1	-1	-1	-1	-1	-1
0	1	-1	-1	-1	-1	1
0	-1	1	-1	-1	-1	-1
0	1	1	-1	-1	-1	-1
0	1	-1	-1	1	-1	1
0	-1	1	-1	1	1	-1
0	1	-1	1	-1	-1	1
0	-1	1	1	-1	1	-1
1	-1	-1	-1	-1	0	0
1	1	-1	1	-1	0	0
1	1	1	-1	1	0	0

A Energia, que faz parte dos neurônios de entrada, e que tem seu valor igual a 0 (zero) é considerada baixa, com isto habilita a rede de controle de energia. Os estados variam dependendo do que se detectou pelas redes sensórias representadas pelas entradas SRSE (Saída da Rede Sensória de Energia) e SRSD (Saída da Rede Sensória de Distância).

As outras duas entradas representam a realimentação da Rede de Controle de Energia, uma vez que ela é recorrente, e são respectivamente RRCE1 (Realimentação 1 da Rede de Controle de Energia) e RRCE2 (Realimentação 2 da Rede de Controle de Energia). As saídas dessa rede representam os estados do AEF e são SRCE1 (Saída 1 da Rede de Controle de Energia) e SRCE2 (Saída 2 da Rede de Controle de Energia).

Também, pelo conjunto de treinamento, podemos ver que nos casos onde o valor da Energia é igual a 1 (um) as saídas da rede é igual a 0, ou seja, a Rede de Controle de Energia está desabilitada e a Rede de Controle Seguir Parede passa a estar habilitada.

5.1.3.5 Conjunto de Treinamento da Rede Motora de Distância

A descrição apresentada na seção 4.2.2.6 pode ser aplicada ao robô e a configuração da hierarquia em relação à Rede Motora de Distância também é plausível. Para este “complicado” controle, desenvolveu-se um conjunto de treinamento com 238 (duzentos e trinta e oito) situações, mas pelo seu tamanho fica inviável mostrá-lo por inteiro.

As situações possíveis dos sensores de distância são repetidas para cada estado da Rede de Controle Seguir Parede e para a Rede de Controle de Energia. Segue abaixo parte do conjunto de treinamento:

Tabela 5.18: Conjunto de Treinamento da Rede Motora de Distância

S1	S2	S3	S4	S5	S6	S7	S8	SRCSP1	SRCSP2	SRCE1	SRCE2	SME	SMD
-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1
1	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	1	1
-1	-1	1	1	-1	-1	-1	-1	-1	-1	0	0	1	-1
1	-1	-1	-1	-1	-1	-1	-1	-1	1	0	0	1	1
-1	-1	1	1	-1	-1	-1	-1	-1	1	0	0	1	-1
-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	1	1
1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	-1	1	-1
-1	-1	1	1	-1	-1	-1	-1	0	0	-1	-1	1	-1
1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	1	1	-1
-1	-1	1	1	-1	-1	-1	-1	0	0	1	-1	1	-1

O conjunto de treinamento acima procura mostrar as diferenças nas atuações da Rede Motora de Distância sobre o controle da Rede de Controle Seguir Parede e sobre o controle da Rede de Controle de Energia. Os 5 (cinco) primeiros conjuntos de dados mostram o controle da Rede de Controle Seguir Parede, neste caso o robô irá sempre em frente quando os sensores não detectarem obstáculo.

Quando os sensores da esquerda, representados por S1 e S2, detectarem algo, o robô irá seguir em frente, mantendo-se próximo desse “objeto” detectado.

E quando os sensores frontais, representados por S3 e S4, detectarem o obstáculo, o robô recuará um motor e avançará o outro no intuito de deixar o obstáculo a sua esquerda, portanto, recuará o motor direito e avançará o motor esquerdo.

Os 5 (cinco) últimos conjuntos de dados referem-se ao controle da Rede de Controle de Energia. Quando o robô estiver com a energia baixa, e não detectar o ponto de energia e nem obstáculos (estado 0 “zero”), os motores do robô sempre avançarão.

Caso algum dos sensores detectar algo, dependendo de qual sensor detectou, um motor avança e o outro recua, no intuito de desviar do obstáculo. Nunca haverá a situação onde os dois motores irão recuar, no entanto, se os sensores traseiros detectarem um obstáculo, os motores avançarão para sair dele.

Os sensores do robô são representados por S_x , onde x varia de 1 a 8, e as saídas da Rede de Controle Seguir Parede estão representadas por SRCSP1 e SRCSP2 (Saída da Rede de Controle Seguir Parede 1 e 2). As saídas da Rede de Controle de Energia estão representadas por SRCE1 e SRCE2 (Saída da Rede de Controle de Energia 1 e 2).

As saídas da Rede Motora de Distância são SMD (Saída para o Motor Direito) e SME (Saída para o Motor Esquerdo). O valor -1 indica que os sensores não estão detectando algo, e o valor 1 significa que aquele sensor detectou algo.

Quando as entradas referentes às saídas das redes de controle estiverem com o valor 0 (zero), a rede está desabilitada e não exerce influência sobre o motor do robô.

5.1.3.6 Conjunto de treinamento da Rede Motora de Energia

Seu conjunto de treinamento é um pouco menor do que o da Rede Motora de Distância, mas continua sendo grande para ser apresentado por inteiro. O total de

situações possíveis é de 201 (duzentos e uma). Parte do conjunto de treinamento é mostrado a seguir:

Tabela 5.19: Conjunto de Treinamento da Rede Motora de Energia

S1	S2	S3	S4	S5	S6	S7	S8	SRCSP1	SRCSP2	SRCE1	SRCE2	SMD	SME
1	1	1	1	1	1	1	1	-1	-1	0	0	0	0
-1	1	1	1	1	1	1	1	-1	1	0	0	0	0
1	-1	1	1	1	1	1	1	1	-1	0	0	0	0
-1	-1	1	1	1	1	1	1	-1	-1	0	0	0	0
1	-1	-1	-1	1	1	1	1	-1	1	0	0	0	0
1	1	1	1	1	1	1	1	0	0	-1	-1	0	0
1	1	-1	1	1	1	1	1	0	0	-1	1	1	1
1	-1	1	1	1	1	1	1	0	0	-1	1	-1	1
1	1	1	1	-1	-1	1	1	0	0	-1	1	1	-1
1	-1	-1	-1	1	1	1	1	0	0	1	-1	0	0

Descrever esse comportamento é um pouco mais complexo do que o da rede anterior, pelo fato de implicar vários fatores como encontrar um ponto de energia atrás de um obstáculo. Este exemplo pode ser visto na última situação da tabela 5.19, no qual o controle é feito pela Rede de Controle de Energia e um ponto de energia foi encontrado, porém o estado da Rede de Controle de Energia é o 2 (dois), o que significa desviar de obstáculo. Para a rede estar nesse estado, os sensores de distância também devem ter detectado algo, com isso, a prioridade é do comportamento desviar do obstáculo e somente depois ir de encontro ao ponto de energia. Nesse caso a Rede de Controle de Energia desabilita a Rede Motora de Energia e habilita a Rede Motora de Distância.

As 5 (cinco) primeiras situações referem-se a essa Rede quando recebe as saídas da Rede de Controle Seguir Parede. As saídas são iguais a 0 (zero) e, conseqüentemente, não há atuação sobre os motores. Entre as outras 5 (cinco), temos a primeira delas, a sexta situação da tabela, onde as saídas também são 0 (zero) e mesmo o controle sendo feito pela Rede de Controle de Energia, ela estará no estado 0 (zero). Logo, o controle está sendo exercido sobre a Rede Motora de Distância e não sobre a Rede Motora de Energia.

Para a Rede Motora de Energia atuar sobre os motores é necessário que o estado da Rede de Controle de Energia seja 1 (um), e isto está sendo mostrado na sétima, oitava e nona situação da tabela. Estas três situações mostram que o robô sempre

procurará se posicionar de frente para o ponto de luz, por exemplo, se o foco for detectado pelos sensores da direita, representados por S5 e S6, o robô irá avançar o motor esquerdo e recuar o motor direito para o ponto ser detectado pelos sensores frontais.

Os sensores do robô são representados por S_x , onde x varia de 1 a 8, e as saídas da Rede de Controle Seguir Parede por SRCSP1 e SRCSP2 (Saída da Rede de Controle Seguir Parede 1 e 2). As saídas da Rede de Controle de Energia estão representadas por SRCE1 e SRCE2 (Saída da Rede de Controle de Energia 1 e 2).

As saídas da Rede Motora de Energia são SMD (Saída para o Motor Direito) e SME (Saída para o Motor Esquerdo). Como se trata dos sensores de energia, o valor 1 indica que os sensores não estão detectando e o valor -1 significa que aquele sensor detectou algo.

Esta inversão com os valores dos sensores acontece pelo fato dos sensores de energia variarem de, aproximadamente, 600 (seiscentos) para ausência do Ponto de Energia até 0 (zero), para Presença do Ponto de Energia.

Quando as entradas referentes às saídas das redes de controle estiverem com o valor 0 (zero), significa que a rede está desabilitada, não exercendo influência sobre os motores do robô.

5.1.4 Seqüência dos Experimentos

Assim como o treinamento das redes da primeira hierarquia foi “off-line”, esse também será. Com os conjuntos de treinamento definidos, passamos para o treinamento, no qual foi usado o algoritmo de “Retropropagação”.

As redes foram treinadas separadas, uma a uma, e seus arquivos de definição e de pesos foram gravados no diretório do Khepera: /SIM, para que no momento da execução do simulador eles sejam lidos.

Os testes, para a atual hierarquia, seguiram os mesmos padrões dos primeiros, no que diz respeito à configuração dos mundos. Primeiro construiu-se um mundo onde só existiam as paredes, com o intuito de analisar o comportamento do robô enquanto a energia estivesse normal e quando estivesse baixa, porém, sem a existência de um ponto de energia para ele reabastecer.

No primeiro teste espera-se que o robô vá ao encontro da parede e fique seguindo-a até atingir o nível de energia baixa. Quando atingir esse nível, espera-se que o robô circule pelo mundo procurando um ponto de energia para reabastecer-se. Como não foi colocado o ponto de energia nesse mundo, ele ficará circulando infinitamente, pois não foi implementado um tempo limite para o robô simulado.

Para o segundo teste criou-se um mundo com as paredes e um ponto de energia. A colocação do ponto, dentro do mundo, foi sendo variada. Espera-se que o robô novamente vá ao encontro da parede e siga-a, enquanto sua energia estiver normal, e passe a percorrer o mundo em busca do ponto de energia, quando sua energia ficar baixa.

O tempo para encontrar um ponto não é importante, mas ao encontrá-lo o robô deve aproximar-se e ficar inerte por alguns segundos, simulando o reabastecimento. Depois de reabastecido deverá ir ao encontro da parede novamente e segui-la, até sua energia voltar a baixar. O robô ficará variando esses comportamentos enquanto o botão "RUN" estiver pressionado ou o simulador estiver ligado.

No terceiro e último teste, configurou-se um mundo com obstáculos e um ponto de energia. A localização dos obstáculos e do ponto de energia foi modificada por várias vezes.

Com essa configuração, mais complexa, poderemos determinar se a hierarquia é mesmo válida ou não, pois o robô deverá seguir a parede e quando sua energia baixar, procurar o ponto de energia, desviando dos obstáculos colocados para atrapalhá-lo.

Depois de reabastecer sua energia, o robô procurará uma parede para segui-la, porém, se encontrar um desses obstáculos pela frente poderá ficar seguindo-o, uma vez que não saberá distinguir o que é parede ou obstáculo.

Pode parecer um comportamento errado, mas não é. Como foi citado o exemplo da barata, na seção 4.2.2 página 64, que ao cruzar o meio de uma sala, encontra um sofá e fica dando voltas em torno dele. O robô terá esse mesmo comportamento, mesmo que não seja um comportamento desejável para o robô.

5.1.5 Análise dos Resultados

Nos dois primeiros testes, verificou-se que o robô seguiu a parede enquanto sua energia estava normal, quando atingiu o nível baixo, passou a percorrer o mundo procurando um ponto de energia.

No primeiro, como não foi implementado um limite de tempo para ele encontrar o ponto de energia, o robô ficou circulando e desviando dos obstáculos (paredes) até o botão “RUN” foi pressionado novamente.

No segundo, o robô seguiu a parede até atingir o nível baixo de energia, então saiu a procura do ponto de energia e quando o encontrou, parou por alguns segundos. Depois que sua energia voltou ao nível normal, ele saiu ao encontro de uma parede, voltando a segui-la, até sua energia baixar novamente.

Assim, ele ficou até o botão “RUN”, no simulador do Khepera, fosse pressionado, causando o fim da execução. Esse teste foi executado várias vezes, modificando-se a posição do ponto de energia e os resultados foram os mesmos.

No último e mais importante teste, configurou-se o mundo de maneira que, num primeiro momento, nenhum obstáculo iria estar entre o robô e a parede. A idéia é que o robô atinja a parede e fique seguindo-a até sua energia baixar e, então, sair a procura do ponto de energia, desviando dos obstáculos que se contraponham entre ele e o ponto.

Num segundo momento, se ele encontrar um obstáculo antes de atingir a parede, ficará seguindo este obstáculo até sua energia baixar. Após encontrar o ponto e reabastecer, sairá atrás de uma “parede” para segui-la, mas poderá encontrar novamente o mesmo, ou outro, obstáculo e ficar seguindo-o.

Foi o que aconteceu, ele desviou de obstáculos quando foi ao encontro do ponto de energia e, após o reabastecimento, seguiu o primeiro objeto detectado pelos sensores de distância, não importando se era “parede” ou “obstáculo”. Por várias vezes o mundo foi modificado, mas o resultado encontrado foi sempre o comportamento descrito acima.

6 CONCLUSÃO

A intensa busca pelo AA perfeito, leva-nos a grandes decepções quando o objetivo não é alcançado. Contudo, esta decepção nunca deve tirar a vontade e determinação em encontrar o caminho para chegar em nossos objetivos.

Durante bilhões de anos a Natureza “iniciou o trabalho” para que existissem seres mais aptos a viver neste ambiente, que podemos considerar hostil, porque, os mais simples seres e até mesmo os mais evoluídos vivem em constante competição.

Vivendo dessa maneira, os seres, quase sempre se encontram em situações de risco, podendo levá-los à morte. Mas, cada espécie desenvolveu seu “mecanismo de segurança”, com isso, o indivíduo de cada espécie pode detectar o que está acontecendo ao seu redor e, assim, poderá tomar atitudes que podem levá-lo a escapar de situações de risco, garantindo a sua sobrevivência e a de sua espécie.

Estes mecanismos são diferentes para cada espécie, por exemplo, a barata tem os cercis (pelinhos na sua parte traseira) que é um mecanismo de detecção de deslocamento do ar, podendo, através deles, sentir a presença de um predador ou de qualquer coisa que possa representar perigo para ela. A espécie Humana é dotada de vários mecanismos como olfato, tato, paladar, etc, que nos ajudam a sentir um perigo eminente ou situações prazerosas.

Na maioria das vezes a base para estes comportamentos, por exemplo, fugir de predador e evitar o perigo, é o SNC. O sistema nervoso, através dos neurônios, recebe informações externas ou internas e, assim, pode “escolher” qual comportamento deve emergir naquele momento. Esta escolha ocorre devido à capacidade do SNC, através dos neurônios, ativar ou desativar um comportamento.

Um exemplo de comportamento é o ataque de apenas uma abelha. Nós não iremos sair correndo desesperados quando apenas uma abelha nos atacar, mas se um cão grande e bravo estiver preparando seu ataque contra nós, com certeza, iremos tomar uma atitude, que na maioria dos casos é sair correndo.

O que estes exemplos querem destacar é que, mesmo estando em uma situação de perigo o comportamento poderá ser diferente, isto é, não iremos correr de uma abelha, mas, iremos correr de um enxame. O mesmo problema, só que com proporções diferentes. Nosso sistema nervoso “diz” que não precisamos correr de uma abelha, mas de um enxame.

Estes comportamentos são característicos dos animais, porém, nós estamos tentando reproduzi-los em seres mecânicos feitos de polias, fios, metais, etc. O que a Natureza levou bilhões de anos para desenvolver, a Ciência está tentando desenvolver em décadas. Já chegamos a resultados esplêndidos, como aqueles citados na introdução deste trabalho, contudo, sempre pensamos em AA's sendo a presa. Podemos pensar neles como sendo predadores, pois na Natureza alguns animais podem ser predadores e presas ao mesmo tempo. Isto nos abrirá outras possibilidades de implementação por serem comportamentos diferentes.

Entramos numa jornada sem vermos seu fim, bem como, estamos buscando freneticamente o AA considerado “perfeito”. Para atingir este objetivo devemos agir como a Natureza, que não criou o ser mais evoluído, entretanto, durante bilhões de anos veio evoluindo esses seres até chegar nos atuais e ainda continua fazendo algumas mudanças.

Para atingir o fim desta busca, foram estudadas as técnicas de IA, algumas podendo ser encontradas em WINSTON (1984), BARRETO (2000). Neste Trabalho apresentamos uma das formas encontrada para dar inteligência a um AA. A rede neural artificial, inspirada na natureza, é uma das ferramentas usadas nessa busca pelo AA inteligente e foi usada por nós para implementar os comportamentos do Robô de Khepera.

Buscando atingir um nível mais elevado na implementação de comportamentos, chegamos a uma arquitetura da rede neural capaz de coordenar e selecionar o comportamento desse robô. As implementações vistas até agora eram ineficazes quando tinham que “tomar decisões”, isto é, escolher o comportamento mais adequado. Uma proposta que tem uma certa vantagem em relação as implementações antigas, é a de BROOKS (1990b), comentada no capítulo 3. Através dela um AA é capaz de ter vários

comportamentos, por exemplo, desviar de obstáculo enquanto tenta atingir um objetivo, como chegar a um canto de uma sala.

Neste exemplo, o AA para chegar ao objetivo deverá superar qualquer obstáculo, sem se chocar ou perder sua rota. Nos momentos em que o AA estiver desviando dos obstáculos, os outros comportamentos estarão sendo inibidos.

Ao longo da dissertação, mostrou-se como o sistema nervoso dos animais se organiza e evolui para gerar comportamentos. E, principalmente, buscou-se mostrar que a organização, do sistema nervoso, é hierárquica e modular. Foram mostradas também, quais as principais alternativas de implementação de agentes autônomos baseados em comportamento, através das propostas de Brooks e de Braitenberg, e que limitações estas propostas apresentavam. Tomando o SNC e as alternativas de implementações como inspiração, sugeriu-se uma nova alternativa, baseada na técnica de redes neurais artificiais, para implementação de comportamentos, em que o ponto principal e diferente é a organização modular e hierárquica, baseada naquela verificada no sistema nervoso.

Se fosse possível olhar esta rede tridimensionalmente, veríamos que no nível mais baixo temos as redes diretas responsáveis pelos comportamentos reflexivos e em um nível superior estão as redes responsáveis pelos comportamentos do tipo reativos.

Esta hierarquia possibilitou a seleção de comportamentos reflexivos através das redes do nível superior. A implementação que fizemos no Khepera tem o comportamento “seguir parede”, “desviar de obstáculos” e “encontrar ponto de Energia”. Cada um, em determinado momento, torna-se prioritário, então, sempre que o robô estiver seguindo a parede, terá esse comportamento como sendo prioritário, caso ele encontre um obstáculo, deverá desviar, passando a prioridade para o comportamento desviar de obstáculos. Quando a energia estiver baixa, a prioridade é do comportamento “encontrar ponto de energia”, porém, se um obstáculo estiver no seu caminho, a prioridade passa a ser do comportamento desviar do obstáculo.

Isto realmente ocorre com o robô. Para todos os testes, a que o robô simulado foi submetido, houve uma resposta, no mínimo, satisfatória. Nos dois primeiros testes, com

a complexidade do mundo baixíssima, o robô seguiu a parede até sua energia ficar baixa e, então, passou a percorrer o mundo em busca de um ponto de energia para seu reabastecimento. No caso do primeiro teste, não havendo o ponto de energia, ele ficou circulando e desviando dos obstáculos (paredes), parando apenas quando “desligado”.

No segundo teste, existia um ponto de energia, ele seguiu a parede e, no momento que deveria reabastecer, encontrou o ponto e parou para reabastecer, voltando depois a seguir a parede. Ficou nessa sequência até que fosse desligado.

O terceiro teste, o principal, não foi diferente dos outros dois, uma vez que o robô seguiu a parede e também, quando precisou, procurou o ponto de energia para reabastecer, desviando dos obstáculos que se contrapunham entre ele o ponto de energia. Assim como nós quando tentamos atingir um objetivo e encontramos algo no caminho que possa nos atrapalhar, procuramos desviar dessa barreira e chegar ao nosso objetivo, o robô também assumiu esta posição, pois primeiro ele desviou dos obstáculos para depois ir ao encontro do ponto de energia, que era seu objetivo.

Este exemplo demonstra que nosso objetivo de implementarmos uma arquitetura para coordenar e selecionar os comportamentos, a princípio, deu um resultado positivo, assim como os outros testes também seriam positivos. Em algumas partes deste trabalho, foram descritos os conjuntos de treinamento das redes, porém, alguns conjuntos ficaram grandes não foi possível apresentar na íntegra. Contudo, não é difícil de defini-los por completo. A definição do conjunto, de cada rede, é feita usando a combinação de todas as situações possíveis das entradas.

Com relação à configuração do mundo, nós tornamos cada vez mais complexo para o robô, na maioria dos casos ele saiu-se bem. Houve casos em que ao desviar do obstáculo, ele não foi ao encontro do ponto de energia, contudo, isto ocorreu devido ao obstáculo ser muito grande, em relação à distância que o robô consegue detectar o ponto de energia, ocasionando a “perda” de contato com o ponto, isto é, seus sensores deixaram de detectá-lo.

O propósito foi alcançado pela Rede Neural Hierárquica. Todos os resultados esperados foram atingidos com satisfação, salvo algumas situações extremas. Algumas

delas foram citadas acima.

Estes comportamentos foram implementados para validar nossa proposta e sabemos que se outros forem implementados, poderemos ter os mesmos resultados. A primeira vista falar que o robô tinha a “função” de ficar vagando sem objetivo pelo mundo, pode parecer estranho, mas muitas vezes nós também andamos sem rumo. A intenção é de mostrar que um robô poderia ficar andando por um determinado local, sem conhecer o percurso, e não se chocar com os obstáculos imprevisíveis que estão no ambiente e que podem atrapalhá-lo.

Nosso limite, para um AA inteligente, reside no fato de que quanto maior a complexidade que queremos dar ao nosso AA, maior será o recurso computacional exigido.

6.1 Proposta Para Trabalhos Futuros

Para trabalhos futuros pode-se implementar, em um nível mais alto da hierarquia, técnicas de IA simbólica, ou outras que sejam eficientes (BARONE et al, 1997) (BARRETO et al, 1997), para um problema que as Redes Neurais não são indicadas, que é o reconhecimento de trajetórias. Também, pode-se buscar alternativas como a criação de redes híbridas com Algoritmos Genéticos (AG) (GOMEZ & MIIKKULAINEN, 1997). Outras técnicas também podem ser empregadas nesta busca, como, otimizar os cálculos para encontrar os pesos entre as conexões (ULIANA et al, 1997).

Uma implementação muito importante a ser feita, para que esta arquitetura fique ainda mais eficiente, é fazer o aprendizado “On-Line”, uma vez que, neste trabalho ele está sendo feito em “Off-Line”.

Para este aprendizado seria ideal buscar alguma técnica biologicamente inspirada. O que não deixa de fora desta técnica os Algoritmos Genéticos. Não é indicado o uso do Algoritmo de “Retropropagação”, mesmo buscando alternativas como a aceleração do algoritmo.(PERA & ESPAIN, 1997)

Os valores definidos para o treinamento foram baseados na idéia de que o robô simulado reage a duas situações extremas: O robô está muito próximo do obstáculo ou não detecta obstáculo e, o robô está muito próximo do ponto de energia ou não o detecta. Podendo haver a existência dos obstáculos ou do ponto de energia, assim como de ambos.

Então, para os testes, os conjuntos de treinamento destas redes foram definidos com valores extremos, isto é, trabalhamos com os dois limites dos sensores, nos quais, para os sensores de distância, 0 significa total ausência de obstáculos e, aproximadamente 1000 (mil), significa choque com o obstáculo. Para os sensores de energia, aproximadamente 600 (seiscentos), significa total ausência de ponto e 0 significa que o robô está muito próximo do ponto de energia. No caso dos sensores de distância, o robô vagava pelo mundo quando seus sensores não detectavam obstáculo, e desviava de obstáculos quando seus sensores detectavam o obstáculo muito perto.

Os sensores de energia também funcionaram dessa forma, se os sensores não detectavam o ponto, os valores eram maiores que 500 (quinhentos), entretanto, se ele detectava o ponto muito próximo, os valores dos sensores eram menores que 50 (cinquenta), então, ele ia em direção ao ponto. Mesmo com esta capacidade de interpolação, o que é considerado bom para esta rede, por estarmos colocando apenas os valores extremos, e com os resultados preliminares e finais sendo de grande importância para este trabalho, teremos que melhorar, tanto os valores dos conjuntos de treinamento, como também, procurar avaliar se esta rede permanecerá com bons resultados após ser inserido mais comportamentos.

Uma sugestão para investigações futuras diz respeito a complexidade da arquitetura hierárquica quando do aumento do repertório de comportamentos, isto é, em que proporção aumentarão o número de neurônios, de interconexões, de redes da hierarquia e do tempo para aprendizado, a medida que se for implementando novos comportamentos.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, J. B. da Mota. **Ficção, realidade e expectativa de robôs inteligentes baseados em comportamento.** In Anais do 1. Simpósio Brasileiro de Automação Inteligente, Rio Claro, SP, 1993. pp 145-154.
- ANDERSEN, Tim L.; MARTINEZ, Tony R. **A Provably Convergent Dynamic Training Method for Multi-layer Perceptron Networks.** Proceedings of the 2nd International Symposium on Neuroinformatics and Neurocomputers, 1995, pp. 77-84.
- ANDERSEN, Tim L.; MARTINEZ, Tony R. **The Effect of Decision Surface Fitness on Dynamic Multi-layer Perceptron Networks (DMP1).** Reference: WCNN, 1996A, pp 177-181.
- ANDERSEN, Tim L.; MARTINEZ, Tony R. **Using Multiple Node Types to Improve the Performance of DMP (Dynamic Multilayer Perceptron).** Proceedings of the IASTED International Conference on Artificial Intelligence, Expert Systems and Neural Networks Reference. 1996B, pp 249-252.
- ANDERSON, Tracy L.; DONATH, M. **Animal Behavior as a Paradigm for Developing Robot Autonomy.** In: Maes, P. (ed.) *Designing Autonomous Agents. Theory and Practice from Biology to Engineering and back.* MIT Press. 1991. p. 145-168.
- DE AZEVEDO, Fernando Mendes. **Uma Proposta de Modelos Formais de Neurônios e Redes Neurais Artificiais.** III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis. Julho 1997, p 503-514.
- BARONE, Dante A. C.; NEWTON, C. Krás Borges; COELHO, Joelson; et al. **Algoritmo Genético no Controle da Trajetória de um Veículo.** III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 402-407.
- BARRETO, Jorge M. **Inteligência Artificial, no limiar do século XXI: Abordagem Híbrida – Simbólica, conexionista e evolutiva.** Florianópolis: ppp Edições, 2000. 2ª ed. 324 p.
- BARRETO, Guilherme de A.; ARAÚJO, Aluizio F. R.; ROSA, Marcelo de O. **Algoritmo de Busca Heurística Usando Redes Neurais Competitivas para Planejamento Ótimo de Trajetória de um Robô Móvel.** III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis. Julho 1997, p 408-413.

- BEER, Randall D., CHIEL, Hillel J., and STERLING, Leon S. **A Biological Perspective on Autonomous Agent Design**. In. Maes, P. (ed.) *Designing Autonomous Agents. Theory and Practice from Biology to Engineering and back*. MIT Press. 1991. p. 169-186.
- BOERS, Egbert; KUIPER, Herman. **Biological metaphors and the design of modular artificial neural networks**. M.Sc. Dissertation, Leiden University. 1992.
- BORGES, G. A.; LIMA, A. M. N. **Sistema Óptico de Reconhecimento de Trajetória para Veículos Autônomos Utilizando Redes Neurais**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis. Julho 1997, p 414-418.
- BRASIL, L. M.; AZEVEDO, F. M. de; BARRETO, Jorge Muniz. **Uma Arquitetura Híbrida Para Sistemas Especialistas**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 167-172.
- BRETON, S.; GRESSER, J.; URBAN, J.P. et al. **Continuous scalar-valued multivariable function learning for inverse problem solving: Application to robotic control**. Technical Report : EEA-TROP-TR-97-08. Faculté des Sciences et Techniques. Mulhouse, France. August 1997.
- BROOKS, Rodney A. **A Robust Layered Control System for a Mobile Robot**. AIM-Memo 864, Massachusetts Institute of Technology – Artificial Intelligence Laboratory. September 1985.
- BROOKS, Rodney A. **Achieving Artificial Intelligence Through Building Robots**. A. I. Memo 899. Massachusetts Institute of Technology – Artificial Intelligence Laboratory. May 1986.
- BROOKS, Rodney A. **The Behavior Language; User's Guide**. A. I. Memo 1227. Massachusetts Institute of Technology – Artificial Intelligence Laboratory. April, 1990a.
- BROOKS, Rodney A. **Elephants Don't Play Chess**. In. Maes, P. (ed.) *Designing Autonomous Agents. Theory and Practice from Biology to Engineering and back*. MIT Press. p. 3-15. 1990b.
- BUESSLER, J. L.; URBAN, J. P. **Visually Guided Movements: Learning With Modular Neural Maps in Robotics**. Technical Report : EEA-TROP-TR-97-10. Faculté des Sciences et Techniques. Mulhouse, France. October 1997.
- CANNY, John. **Collision Detection for Moving Polyhedra**. A. I. Memo 806. Massachusetts Institute of Technology – Artificial Intelligence Laboratory. October 1984.

CARPENTER, Malcolm, B. **Neuroanatomia Humana**. Rio de Janeiro: Interamericana, 1978. 7ª edição. 700 p.

CARPINTEIRO, Otávio A. S. **A hierarchical self-organizing map model for pattern recognition**. . III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis. Julho 1997, p 484-488.

CONTU, Paolo; OSÓRIO, Paulo A. **Anatomia Funcional do Sistema Nervoso**. Porto Alegre: Edições URGs. 1972. 340p.

CHURCHLAND, Paul M. **Matter and Conciousness: a contemporary introduction to the philosophy of mind**. Cambridge, Massachusetts, A Bradford Book. The MIT Press, 1984, second edition. 180 p.

CRISTIAN, Andrew Dean. **Design and Implementation of a Flexible Robot**. Massachusetts Institute of Technology. Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degree of Science in Mechanical Engineering. August 1989.

DAWKINS, Marian Stamp. **Explicando o comportamento Animal**. Manole Ltda. São Paulo. 1989.

ERHART, Eros Abrantes. **Neuroanatomia**. São Paulo: Ateneu. 1974. 5ª ed. 420p.

FALQUETO, J.; BARRETO, J. M.; BORGES, P.S.S. et al. The Measurement of Artificial Intelligence: An IQ for Machines?. Proceedings of the IASTED International Conference on Modelling, Simulation and Control Innsbruck, Austria, February 19-22, 2001. Vol I, pp 409-413, Acta Press, Anaheim Calgary Zurich, ISBN 0-888986-316-4; ISSN 1025-8973.

FERRELL, Cynthia. **Robust Agent Control of an Autonomous Robot with Many Sensors and Actuators**. Massachusetts Institute of Technology. A revised version of the thesis submitted in partial fulfillment of the requirements of the degree of Master of Science in Electrical Engineering and Computer Science. May 1993. ferrel@ai.mit.edu.

FRANDSON, R.D. **Anatomia e Fisiologia dos Animais Domésticos**. Ed. Guanabara Koogan SA. Segunda Edição. Rio de Janeiro. 1979. 429p.

FREEMAN, James A., SKAPURA, David M. **Neural Networks: Algorithms, Applications and Programming Techniques**. Ed. Addison-Wesley Publishing Company. July. 1992. 401p.

GANONG, W.F. **Fisiologia Médica**. Ed Atheneu. 5ª Edição. São Paulo. 1985. 680p.

GETTY, Robert. **Anatomia dos Animais Domésticos**. Ed. Guanabara Koogan SA. Vol 1. 5ª Edição. Rio de Janeiro. 1986. 1134p.

- GOMEZ, Faustino; MIIKKULAINEN, Risto. **Incremental Evolution of Complex General Behavior**. To appear in *Adaptive Behavior*, Department of Computer Sciences, The University of Texas at Austin. 1997. pp 317-342.
- GUYTON, Arthur C. **Fisiologia Humana**. Rio de Janeiro: Interamericana. 1980. 4ª edição. 492 p.
- HANCOCK, John A. **Laser Intensity-Based Obstacle Detection and Tracking**. Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy. The Robotics Institute Carnegie Mellon University Pittsburgh. Pennsylvania. January 1999. 180 p.
- HILLER, Martha Jean. **The Role of Chemical Mechanisms in Neural Computation and Learning**. Massachusetts Institute of Technology. Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy. May 1993.
- HOGG, David W.; MARTIN, Fred; RESNIK, Mitchel. **Braitenberg Creatures**. June 1991
- KOVÁCS, Zsolt L. **O Cérebro e a Sua Mente: Uma Introdução à Neurociência Computacional**. São Paulo: Edição Acadêmica, 1997. 214 p.
- MAES, Pattie. **Situated Agents Can Have Goals**. Guest Editorial. In. Maes, P. (ed.). *Designing Autonomous Agents. Theory and Practice from Biology to Engineering and back*. MIT Press. 1990. p 49-70.
- MAES, Pattie. **Modeling Adaptive Autonomous Agents**. Massachusetts Institute of Technology –MIT. Cambridge, Massachusetts. 1995.
- MATARIC, Maja J. **Interaction and Intelligent Behavior**. Massachusetts Institute of Technology. Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy. May 1994.
- MCFARLAND, David; BÖSSER, Thomas. **Intelligent Behavior in Animals and Robots**. Massachusetts Institute of Technology –MIT. Cambridge, Massachusetts. 1993. 305p.
- MEDEIROS, Adelardo A. D. **A Survey of Control Architectures for Autonomous Mobile Robots**. *Journal of the Brazilian Computer Society*. V. 4, n. 3 Campinas Abril, 1998.
- MERLAT, Lionel; KÖNEKE, Axel; MERCKLÉ, Jean. **A Tutorial Introduction to Cellular Neural Networks**. Intern Workshop in Electrical Engineering and Automatic Control, ESSAIM, Mulhouse, France. March 1997.

MESSENGER, J. B. **Nervos, cérebro e comportamento**. Coleção Temas de Biologia Vol. 22, Epu: ed. da Univers. de São Paulo. São Paulo. 1980.

MORIARTY David E.; MIKKULAINEN, Risto. **Hierarchical Evolution of Neural Networks**. In Proceedings of the 1998 IEEE Conference on Evolutionary Computation (ICEC-98, Anchorage, AK). Piscataway, NJ: IEEE, 1998.

NABBE, Bart. **A Language for Reconfigurable Robot Control**. Submitted to the Department of Computer Science in partial fulfillment of the requirements for the degree of Master of Science. University of Amsterdam. Amsterdam. September 1998. 59 p.

NASCIMENTO, Paulo C.; SILVA, Alexandre P. Alves da. **Self-Organizing Neural Network 2**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 45-50.

NETO, Areolino de Almeida; GOES, Luiz Carlos Sandoval. **Projeto de um Controlador Neural para o Robô ITA-IEMP**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 384-389.

NIYOGI, Partha. **The Informational Complexity of Learning from Examples**. Massachusetts Institute of Technology. Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy. February 1995.

OLIVEIRA, Jurandir Paz de; ULIANA, Policarpo Batista; LIMA, Walter Celso. **Algoritmos de Extração de Regras de Redes Neurais Artificiais**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p173-178.

ORGEL, Leslie E. **As Origens da Vida**. Brasília: Editora Universidade de Brasília, 1973. 195p.

PERA, Vitor; ESPAIN, Carlos. **Aceleração do Algoritmo de Backpropagation**. III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 33-38.

ROISENBERG, Mauro; BARRETO, J. M.; AZEVEDO, F. M de. **Biological Inspirations in Neural Networks Implementations of Autonomous Agents**. SBIA. Curitiba. 1996.

ROISENBERG, Mauro. **Emergência da Inteligência em Agentes Autônomos através de Modelos Inspirados na Natureza**. Tese Submetida à Universidade Federal de Santa Catarina Como Parte dos Requisitos para a Obtenção do Grau de Doutor em Engenharia Elétrica – Área de Concentração em Sistemas de Informação. Universidade Federal de Santa Catarina - UFSC. Florianópolis 1998. 206 p.

ROISENBERG, Mauro; BARRETO, J. M.; AZEVEDO, F. M de. **Um Ambiente Evolucionário para Geração de Redes Neurais em Agentes Autônomos.** Proceedings of IV Brazilian Conference on Neural Networks - IV Congresso Brasileiro de Redes Neurais, ITA, São José dos Campos - SP - Brazil, pp. 888-999, Julho 1999.

SAKSIDA, Lisa Marie. **The Interaction of Perception and Cognition: A Competitive Connectionist Model of the Effects of Experience on Perceptual Representations.** A Dissertation Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy. Robotics Institute and Center for the Neural Basis of Cognition. School of Computer Science Carnegie Mellon University Pittsburgh. Pennsylvania. August 1999. 184 p.

SARACHIK, Karen Beth. **Visual Navigation: Constructing and Utilizing Simple Maps of an Indoor Environment.** Massachusetts Institute of Technology. Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Master Science. March 1989.

SCHMIDT-NIELSEN, Knut. **Fisiologia Animal.** Ed. Edgard Blücher. São Paulo. 1976. 139p.

SCHMIDT-NIELSEN, Knut. **Fisiologia Animal: Adaptação e Meio Ambiente.** Ed. Livraria Santos Editora. 5ª Edição. São Paulo. 1996. 600p.

SMIEJA, F. J. **Multiple Network Systems (Minos) Module: Task Division and Module Discrimination.** Proceedings of the 8th AISB conference on Artificial Intelligence, Leeds, April 1991.

SMIEJA, F. J.; MÜHLENBEIN, H. **Reflective Modular Neural Networks Systems.** German National Research Center of Computer Science (GMD). GMD Report 633. March 1992.

ULIANA, Policarpo B.; SEARA, Rui; BERMUDEZ, José Carlos M. **Inicialização Não Aleatória de Pesos de Redes Neurais Artificiais com Baixa Complexidade Computacional.** III Congresso Brasileiro de Redes Neurais. IV Escola de Redes Neurais. Florianópolis, julho 1997, p 231-237.

URBAN, J.P.; BUESSLER, J.L.; GRESSER, J. **NeuroModule-Based Visual Servoing of a Robot Arm with a 2 d.o.f. Camera.** Technical Report : EEA-TROP-TR-97-13. Faculté des Sciences et Techniques. Mulhouse, France. October 1997.

WALD, George. **The Origin of Life.** In Hanavalt, P. C. & Haynes R. H. (eds.) **The Chemical Basis of Life: An Introduction to Molecular and Cell Biology.** W. H. Freeman and Co.. San Francisco. 1973. pp.9-17.

WINSTON, Patrick Henry. **Artificial Intelligence**. Massachusetts Addison-Wesley Publishing Company. 1984. Second edition. 524 p.

WOOD, Dennis W. **Princípios de fisiologia animal**. Ed. Polígono. São Paulo. 1973. 366p.

www.egyptinamyths.com. Site sobre o Deus-Sol Re e sua forma Khepera. Último Acesso 04/2001.

ANEXO A - O AMBIENTE DE DESENVOLVIMENTO: “KHEPERA SIMULATOR”

Como já mencionado, para validar nossa proposta usaremos um simulador chamado “Khepera Simulator” ou “Simulador do Robô de Khepera”.

O nome Khepera foi herdado do Deus-Sol Re. Os egípcios, no antigo Egito, via um pequeno escaravelho rolar uma bola de estrume e depositá-la em um buraco (toca) deixando-a lá. Nesta bola de estrume, o escaravelho, botava seus ovos e assim que eram chocados, as larvas comiam a bola de estrume.

Quando a bola acabava, os escaravelhos jovens saiam da toca à procura de alimento. Vendo isto acontecer, os egípcios, acreditavam que os escaravelhos se autogeravam e se auto-regeneravam.

Este fato levou os egípcios a representarem o Deus-Sol Re através de um corpo humano com a cabeça do escaravelho ou com o corpo do escaravelho e a cabeça do Deus-Sol Re, pois eles acreditavam que seu deus tinha a capacidade de se autogerar e se auto-regenerar. O Deus-Sol Re, para os egípcios, rolava o sol através do céu e as vezes também era mostrado rolando a lua, assim como o escaravelho rolava a bola de estrume. (www.egyptianmyths.com, 2001).



Figura A.1: Representação do Deus-Sol Re através da forma “Khepera” (Khepra, Khepri, Khopri)

Então, acreditando que pela aparência do escaravelho Scarab (Kheper), pequeno e arredondado, e pela representação do Deus-Sol Re através da forma “Khepera”, deu-se o nome ao robô “Khepera”.

Neste trabalho, o aperfeiçoamento da hierarquia depende exclusivamente dos resultados alcançados dentro do simulador. Mesmo não tendo o robô real para avaliarmos o desempenho da rede hierárquica em ambientes extremamente “hostis” a ele, temos a certeza de que os resultados que alcançamos com o simulador podem ser refletidos no robô real.

A facilidade para fazer alterações no código do robô, assim como, a facilidade para utilizar a linguagem de programação C ou C++, foi ponto culminante para a escolha do simulador como plataforma para a validação da rede hierárquica.

O pacote permite escrever algoritmos de controle (rede neural, sistema classificador), usando linguagem C ou C++. Uma biblioteca de funções é fornecida para que permita dirigir o robô e exibir os resultados.

O simulador roda no sistema operacional Unix/Linux, em um ambiente X (gráfico). Alguns exemplos de controladores já estão inclusos dentro do pacote, inclusive um controlador de rede neural.

O simulador caracteriza-se também pela possibilidade de transferir o código implementado para o robô real de Khepera, assim você pode muito facilmente transferir seus resultados da simulação para o robô real.

A tela do Simulador de Khepera é dividida em duas partes: o “Mundo” situado na parte esquerda, enquanto o “robô” está à direita. Na parte do mundo, podemos observar o comportamento do robô em seu ambiente considerando que na parte de robô, podemos observar o que está situado na parte interna (sensor, motores e controlador).

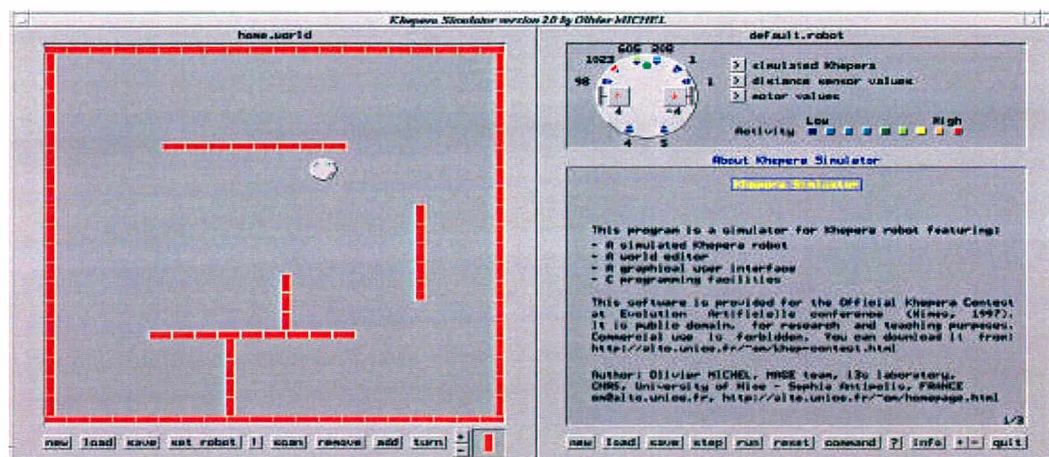


Figura A.2: Ambiente do robô simulado – Robô de Khepera

A.1 Descrição do Mundo

Vários mundos estão disponíveis no diretório SIM/WORLD/. Podemos carregar qualquer mundo (com extensão “world”) pressionando o botão “Load”. Além disso, é possível projetar qualquer mundo novo apenas apertando o botão “New”. Podemos armazenar estes mundos criados usando o botão “Save”.

Estão disponíveis, para acrescentar ao mundo, objetos do tipo tijolos, cortiças (rolhas) ou luminárias que poderão resultar em mundos mais complexos e para isto usamos o botão “Add”. As dimensões deste ambiente simulador (comparado com o robô real Khepera) são 1m - 1m.

Para adicionar ou retirar objetos, utiliza-se os botões “+” e “-”. No caso dos “tijolos” podemos mudar o ângulo de inserção, deixando-o na vertical, horizontal ou diagonal, utilizando o botão “turn” tantas vezes quanto for necessário.

Então, para fixar um objeto em uma localização no mundo, utiliza-se o botão “add”. leve o mouse até a posição e pressione o botão direito. Se quiser remover objetos utiliza-se o botão “Remove”, na interface gráfica, clicando como o botão esquerdo do mouse no objeto que você quer remover. Uma vez colocados objetos, ou após criar um novo mundo, é necessário utilizar o botão “Scan” antes de colocar o robô para “andar”.

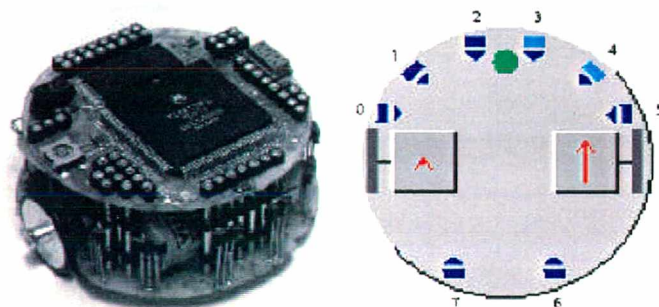


Figura A.3: Khepera (5 cm diâmetro) e sua contraparte simulada

A.2 Descrição do Robô

A.2.1 Apresentação

O robô móvel simulado inclui 8 sensores infravermelhos que permitem descobrir por reflexão (retângulos pequenos) a proximidade de objetos na sua frente, em sua traseira, à direita e à esquerda. Cada sensor devolve um valor que varia entre 0 e 1023, representados por níveis de cores. O 0 (zero) indica que nenhum objeto é percebido, enquanto 1023 indica que o robô já tocou no objeto.

Valores intermediários podem dar uma idéia aproximada da distância entre o sensor e o objeto. Estes sensores também podem medir o nível de luz no ambiente (triângulos pequenos) ao redor do robô. Eles devolvem um valor exibido em níveis de cores, sendo que o valor perto de 500 significa luz ambiente e de 50 significa presença de uma fonte luminosa. Cada motor pode ter um valor de velocidade entre -10 e +10. Setas vermelhas nos motores indicam a velocidade e a direção da mesma.

A.2.2 Os Motores

O robô move-se, de acordo com a velocidade fixada pelo usuário. Um ruído imposto de $\pm 10\%$ é somado ao total da velocidade do motor, enquanto um ruído imposto de $\pm 5\%$ é somado à direção que é o resultado da diferença das velocidades dos motores.

A.2.3 Os sensores

Para calcular a saída do valor de distância, um sensor explora um grupo de 15 pontos de um triângulo na sua frente. Um valor de saída é computado como uma função da presença (ou da ausência) de obstáculos nestes pontos.

Um ruído imposto que corresponde a $\pm 10\%$ do seu total é somado à saída deste valor de distância. O Valor de saída de luminosidade é computada adequadamente para a distância e para o ângulo entre o sensor e a fonte de luz. Um ruído imposto de $\pm 5\%$ é somado a este valor.

A.2.4 Operando o Robô

Para colocar o robô em uma determinada localização no mundo, utiliza-se o botão “Set robot” situado na parte do “mundo”. Com o botão esquerdo do mouse clica-se em uma posição em algum lugar no mundo (não colocar em cima de um objeto). Para colocá-lo em uma direção escolhida por nós, utiliza-se o botão “command” e digita-se o ângulo desejado, o que fará com que o robô gire para alcançar a posição digitada em graus.

Para colocar o robô simulado em funcionamento, basta utilizar o botão “Run” e para parar também. Para movimentar o robô passo a passo, deve-se utilizar o botão “step”. O botão “?” permite testar os sensores do robô (especialmente para o robô real).

A.2.5 Programação

A.2.5.1 Os códigos Fonte do Simulador

Os códigos fonte do simulador estão no diretório SIM/SRC. Eles não devem ser modificados. Isto é muito importante para atualizações adicionais do software. Estas fontes são escritas em ANSI C. Assim, se quisermos escrever nosso controlador em C, é necessário compilar estas fontes com um compilador C (gcc -c), e se preferirmos C++, também podemos compilar estas fontes com um compilador de C++ (substituindo gcc -

c por `g++ -c` no makefile).

A.2.5.2 Arquivos USER

O Arquivo “.simrc” é um arquivo especial relativo a configurações de hardware. É lido através de Simulador de Khepera toda vez que “sim” é executado. É um arquivo oculto no diretório /SIM, dessa forma, é necessário digitar algo como: `ls -a`, para vê-lo. Contém 3 parâmetros importantes que podemos editar:

_ KHEPERA AVAILABLE: pode ser TRUE ou FALSE, dependendo do robô real de Khepera estar conectado ou não.

_ SERIAL PORT: dispositivo da SERIAL na qual o robô é conectado (se disponível). Poderia ser /dev/ttya que representa a PORTA SERIAL A em uma workstation Sun, mas geralmente depende do tipo de computador que estamos usando. Este valor só é usado quando KHEPERA AVAILABLE for TRUE.

_ MONODISPLAY: pode ser TRUE ou FALSE de acordo com o tipo de tela usado. Fixando para TRUE, permite rodar o Simulador de Khepera em uma exibição monocromática.

Quando executamos “sim”, a opção de flag `-s` permite executar o Simulador de Khepera em modo “somente simulação” (não faz uso do vínculo consecutivo para o Khepera real) até mesmo se KHEPERA AVAILABLE for TRUE.

A.2.5.3 Arquivos Fonte do Controlador

Pode-se modificar do modo que se queira. Também é possível adicionar arquivos novos (e por conseguinte modificar o makefile associado). Os arquivos devem ser armazenados em um “diretório” chamado USER, obedecendo ao seguinte caminho (path): SIM/USER. Arquivos básicos vazios do controlador estão disponíveis em um “diretório” chamado EXEMPLE 0, obedecendo ao seguinte caminho:

SIM/EXAMPLES/EXAMPLE 0. Mas, pode-se construir o próprio controlador, devendo começar pelo exemplo 1 que é o controlador padrão do pacote (também disponível no diretório EXAMPLE 1, caminho: SIM/EXAMPLES/EXAMPLE1).

A.2.5.4 Informações Criadas por Usuários

Existe a necessidade de uma área para exibir algumas variáveis, resultados, gráficos, explicações, etc. Esta área existe e lhe permite escrever valores numéricos, texto, desenhos, etc., na janela principal do Simulador de Khepera.

Toda a informação que se pode escrever é dividida em páginas e armazenada em um arquivo e este em um diretório. Um arquivo padrão de três páginas contém uma descrição do Simulador de Khepera. Pode-se trocar o arquivo utilizando o botão “info” e pode-se olhar as páginas utilizando os botões “+” ou “-” próximo ao botão “info”.

Podemos criar até 4 diretórios de informação, cada um contendo um arquivo com até 255 páginas. Para definir isto, teremos que editar o arquivo info.h que está no diretório USER. Depois, teremos que editar a função do Robô - DrawUserInfo(struct Robot *robot, char info, char page), onde “info” é o diretório de informação e “page” é a página atual do arquivo. Aqui estão as constantes a serem editadas no USER, no arquivo info.h header:

_ NUMBER DE INFO é o número de diretórios de informações de usuário. Este valor deve estar entre 0 e 4.

_ PAGES INFO x é o número de páginas para o número de diretório x (x varia de 1 a 4). Estes valores devem estar entre 0 e 255.

_ TITLE INFO x é o título do diretório x (x varia de 1 a 4).

A.2.5.5 Exemplos

Quatro exemplos são encontrados dentro do simulador do Khepera. Eles estão localizados nos diretórios EXAMPLE's, que estão no caminho SIM/EXAMPLES/, onde

cada “diretório” tem o nome de EXEMPLEx, onde x varia de 0 a 3, número de exemplos. Exemplo 0 não contém realmente um exemplo de controlador, mesmo contendo todas as funções necessárias para rodar o Simulador de Khepera, pois elas estão vazias.

Aqui é dada uma descrição pequena dos exemplos:

_ Exemplo 0: o controlador “vazio”.

_ Exemplo 1: um exemplo de algoritmo de controle simples.

_ Exemplo 2: Rede de neural e gnuplot. Este exemplo mostra como implementar rede neural artificial para controlar o robô. Também caracteriza um emissor/receptor para “gnuplot” com utilidade para exibir gráficos. As redes neurais mostradas aqui são resultado de um processo evolutivo que usa algoritmos genéticos, morphogenesis, e metabolismo artificial.

_ Exemplo 3: Módulo Simulador do Khepera multi-agentes, desenvolvido por Manuel Clergue, permite controlar vários robôs simulados do Khepera.

_ Exemplo 4: Simulador do Khepera simula um módulo de dispositivo consecutivo. Este módulo é especialmente útil se estiver desenvolvido um programa que envia comandos a um Khepera real pela Porta Serial de seu computador. Será necessário redirecionar o fluxo de entrada e saída, para o Simulador do Khepera transportar arquivos. Poderá observar o Khepera ser dirigido por seu comando Serial.

A.2.5.6 Programando um Controlador do Robô de Khepera

Para finalizar, mostraremos nesta seção como podemos implementar um algoritmo de controle. Esta implementação é muito simples e foi inspirada em Braitenberg.

Em primeiro lugar, definimos nosso algoritmo:

Repita

Se o robô não perceber nenhum obstáculo, então, mova adiante.

Se um obstáculo é percebido no lado esquerdo do robô, então, vire à direita.

Se um obstáculo é percebido no lado direito do robô, então, vire à esquerda.

Até que seja sentido na parte de trás do robô.

Para programar isto, no Simulador do Khepera, nós precisamos traduzir “não encontrou nenhum obstáculo” ou “se o robô não perceber nenhum obstáculo” em algo que os sensores do robô entenda. Os sensores do robô são legíveis pela variável (Type struct Robot).

O valor que corresponde à medida de distância do sensor 2 dianteiro, por exemplo, é armazenado na equação (6), abaixo descrita:

$$\text{robot->IRSensor}[2].\text{DistanceValue (Type int).} \quad (6)$$

Os valores variam entre 0 e 1023. Assim se este valor exceder um determinado patamar, digamos 950 (que será definido em COLLISION TH constant), podemos considerar que um obstáculo foi sentido por este sensor e está muito próximo.

Para dirigir os motores do robô, tem-se que escrever na variável do robô os valores que correspondem à velocidade de cada motor que queremos aplicar. Estes valores, sendo inteiros, serão escritos na equação (7), descrita a seguir:

$$\text{robot->Motor}[\text{LEFT}].\text{Value and robot->Motor}[\text{RIGHT}].\text{Value} \quad (7)$$

Eles variam entre -10 e +10. Nós definiremos como constantes valores para VELOCIDADE DE RECUO e VELOCIDADE DE AVANÇO.

Todas as entradas e operações de saída têm que acontecer dentro da função boolean do Robô - StepRobot(struct *robot). Esta função devolve FALSE para parar e TRUE caso contrário:

```

#define FORWARD_SPEED 5 - normal (lento) velocidade de avanço;
#define TURN_SPEED 4 - normal (lento) velocidade de recuo;
#define COLLISION_TH 900 - valor do sensor de IR para ser considerado como
colisão;
Boolean StepRobot(struct Robot *robot)
{
If ((robot->IRSensor[0].DistanceValue > COLLISION_TH) || /* esquerda dianteira */
    (Robot->IRSensor[1].DistanceValue > COLLISION_TH) || /* sensor */
    (Robot->IRSensor[2].DistanceValue > COLLISION_TH)) /* se há uma colisão no
lado esquerdo do robô */
{
    Robot->Motor[LEFT].Value = TURN_SPEED;
    Robot->Motor[RIGHT].Value = -TURN_SPEED; /* direito recua */
}
else if ((robot->IRSensor[3].DistanceValue > COLLISION_TH) ||
    (Robot->IRSensor[4].DistanceValue > COLLISION_TH) ||
    (Robot->IRSensor[5].DistanceValue > COLLISION_TH))
    /* se há uma colisão no lado direito do robô */
    {
        Robot->Motor[LEFT].Value = -TURN_SPEED;
        Robot->Motor[RIGHT].Value = TURN_SPEED; /* esquerdo recua */
    }
Else
    {
        Robot->Motor[LEFT].Value = FORWARD_SPEED;
        Robot->Motor[RIGHT].Value = FORWARD_SPEED; /* vai adiante
(padrão) */
    }
If ((robot->IRSensor[6].DistanceValue > COLLISION_TH) || /* colisão na */
    (Robot->IRSensor[7].DistanceValue > COLLISION_TH)) /* parte de trás */
    return(FALSE); /* parada */

```

Else

```
return(TRUE); /* continua */  
}
```

Com este código simples, nós definimos um controlador completo do robô.

Para exibir algum texto, valores numéricos, ou desenhos na área de informação do usuário, usamos a função `DrawUserInfo (...)` no código fonte `user.c` que está no “diretório” `USER`, no caminho `SIM/EXAMPLE1/USER/`.